Deutsches Zentrum
für Luft- und Raumfahrt e.V.

Forschungsbericht 2003-09

Improved Robustness for Numerical Simulation of Turbulent Flows around Civil Transport Aircraft at Flight Reynolds Numbers

Jens K. Fassbender

Institute of Aerodynamics and
Flow Technology
Braunschweig

DLR

Improved Robustness for Numerical Simulation of Turbulent Flows around Civil Transport Aircraft at Flight Reynolds Numbers

Jens K. Fassbender

Institute of Aerodynamics and Flow Technology
Braunschweig

266 Seiten
189 Bilder
21 Tabellen
134 Literaturstellen

Improved Robustness for Numerical Simulation of Turbulent Flows around Civil Transport Aircraft at Flight Reynolds Numbers

Deutsches Zentrum
für Luft- und Raumfahrt e.V.

Abstract

This work aims at providing numerical methods that enable the robust and efficient simulation of turbulent flows around civil transport aircraft configurations at flight Reynolds numbers. The combined utilization of a computational and a theoretical approach for the systematic investigation of the numerical treatment of turbulence equations characterizes the work at hand.
Robustness problems in case of multigrid treatment of advanced transport equation turbulence models have been identified to prevent convergence of simulations at high or flight Reynolds numbers. Therefore the application of multigrid to the turbulence equations is omitted while the multigrid treatment of the RANS equations stays unchanged. Moreover, a fully implicit time integration scheme — a DDADI approach — is applied to the turbulence equations.
This approach was inspired by a systematic investigation of the numerical treatment of turbulence equations utilizing two dimensional computations and Fourier analyses. This study clearly demonstrated a destabilizing effect of productive (turbulence) source terms which is amplified in the framework of multigrid.
Finally, both the original and the improved FLOWer code has been applied to two and three dimensional test cases of industrial relevance and the respective results are compared. The new approach raised robustness to a sufficient niveau to converge viscous computations at flight Reynolds numbers. Furthermore, the convergence speed of aerodynamic coefficients for three dimensional applications has also been improved significantly.

Zusammenfassung

Zielsetzung dieser Arbeit ist die Bereitstellung numerischer Verfahren, die eine robuste und effiziente Simulation turbulenter Strömungen für zivile Transportflugzeugskonfigurationen bei Flug–Reynoldszahlen ermöglichen. Diese Arbeit zeichnet sich dadurch aus, dass sie für die systematische Untersuchung der numerischen Behandlung von Turbulenzgleichungen den Einsatz von Simulationsrechnungen und theoretischen Ansätzen kombiniert.
Robustheitsprobleme, welche die Konvergenz von Simulationen bei hohen und bei Flug–Reynoldszahlen verhindern, konnten auf die Anwendung eines Mehrgitter–Verfahrens auf Transportgleichungsturbulenzmodelle zurückgeführt werden. Daher ist das Mehrgitter–Verfahren für die Turbulenzgleichungen abgeschaltet worden, während sein Einsatz für die RANS Gleichungen nicht verändert wurde. Desweiteren ist ein voll implizites Zeitintegrationsverfahren — ein DDADI Ansatz — auf die Turbulenzgleichungen angewendet worden.
Die Herleitung dieses Ansatzes basiert auf einer systematischen Untersuchung der numerischen Behandlung von Turbulenzgleichungen. Hierfür wurden zwei–dimensionale Simulationsrechnungen und Fourier–Analysen durchgeführt. Die Studie hat deutlich einen destabilisierenden Effekt produktiver (Turbulenz–) Quellterme gezeigt, der im Rahmen von Mehrgitter–Verfahren verstärkt wird.
Abschließend wurden sowohl der originale FLOWer–Code als auch die verbesserte Variante auf zwei– und drei–dimensionale industriell relevante Testfälle angewendet und die Ergebnisse verglichen. Der neue Ansatz liefert eine deutliche Steigerung der Robustheit, aufgrund derer viskose Simulationsrechnungen bei Flug–Reynoldszahlen konvergieren. Darüberhinaus ist die Konvergenz der aerodynamischen Koeffizienten für dreidimensionale Anwendungen deutlich beschleunigt worden.

Vom Fachbereich für Mathematik und Informatik

der Technischen Universität Braunschweig

genehmigte Dissertation

zur Erlangung des Grades

eines Doktor–Ingenieurs (Dr.–Ing.)

Autor: **Dipl.–Math. Jens K. Fassbender**

Titel: **Improved Robustness for Numerical Simulation of Turbulent Flows around Civil Transport Aircraft at Flight Reynolds Numbers**

Eingereicht am	19. Mai 2003
Mündliche Prüfung am	22. August 2003
1. Referent	Prof. H. G. Matthies (PhD)
2. Referent	Prof. Dr.–Ing. habil. C.–C. Rossow
3. Referent	Prof. E. Turkel (PhD)

Preface

The work on hand has been prepared while I have been working as a research assistant and later on as a research scientist at the Institute of Aerodynamics and Flow Technology of DLR, the German Aerospace Center, in Braunschweig, Germany.

I would like to thank Prof. Matthies, Institute of Scientific Computing at the Technical University Braunschweig, Germany, for the supervision of this doctoral thesis and for all his questions on implicitness.
Special thanks goes to Prof. Rossow, Institute of Aerodynamics and Flow Technology of DLR, Braunschweig, Germany, and to Prof. Turkel, School of Mathematical Sciences, Tel Aviv University, Israel, for their work as referees of this doctoral thesis and many fruitful discussions.
I owe Prof. Kroll, Institute of Aerodynamics and Flow Technology of DLR, Braunschweig, Germany, the opportunity to prepare this thesis by providing the topic, the position and the computer resources. Furthermore, I thank Prof. Kroll for all accompanying discussions and for reviewing this work.

I would like to thank my colleagues in the department Numerical Methods and at the Institute of Aerodynamics and Flow Technology (former Institute of Design Aerodynamics) for answering all my questions on (computational) fluid mechanics, the FLOWer code and on the how–tos of MegaCads as well as for replying to my improvements to the FLOWer code.

Thanks I owe to all my friends who believed in me and my saying "…it's nearly finished …" and who prayed for me. Special thanks I have to give to Dr. M. Petri and G. Einnarson for reading and correcting my English.

Last but not least it has to be pointed out that this work would not have been possible without the support of my family. With all my heart I thank my beloved wife for the price she was willing to pay for this thesis.

> My son, there is something else to watch out for. There is no end to the writing of books, and too much study will wear you out.
>
> (Ecclesiastes 12,12; Good News Bible)

Contents

Nomenclature

a	speed of sound
B	implicit smoothing operator
$\mathcal{B}$	Fourier symbol of implicit smoothing operator B
c_p	specific heat coefficient at constant pressure
c_v	specific heat coefficient at constant volume
C	cell face of control cell V
$\mathcal{C}$	area of cell face C
∂V	boundary surface of control cell V
$\partial\mathcal{V}$	area of control cell surface ∂V
D	main diagonal of matrix M
e	inner energy per unit mass
E	total energy per unit mass
$\mathcal{E}_*^*$	index shift operator (subscript: index to be shifted; superscript: shift)
F_*	(operator of) forcing function
$\mathcal{F}^*$	Fourier symbol of (operator of) forcing function F_*
g	amplification factor of time stepping scheme operator $\mathcal{G}$
$\mathcal{G}$	time stepping scheme operator
H	total enthalpy per unit mass / stagnation enthalpy per unit mass
i, j	coordinates of computational domain, i. e. nodal point indices
$\mathbf{I}$	imaginary unit ($= \sqrt{-1}$)
I	identity matrix
I_*^*	transfer operator (subscript: target level; superscript: source level)
$\tilde{I}_*^*$	injection operator (subscript: target level; superscript: source level)
$\mathcal{I}$	identity operator
`k`	coefficient of thermal conductivity
k	turbulent kinetic energy
L	lower part of matrix M
`levels`	number of multigrid levels in use
$\lim_*$	limiting factor for turbulence variables (denoted by subscript)
M	Mach number
M	matrix of linearized implicit system of equations
M	operator related to matrix M
$\mathcal{M}$	Fourier symbol of operator M
$\vec{n}$	outer normal unit vector
$\mathcal{P}_*^*$	Fourier symbol of prolongation operator
q	number of stages in Runge–Kutta step
$q_{\vec{n}}$	velocity in direction of $\vec{n}$, i. e. $\vec{q} \bullet \vec{n}$
$\vec{q}$	velocity vector

R	residual
$\widetilde{R}$	residual on coarse grids modified by forcing function
$\mathcal{R}_*^*$	Fourier symbol of restriction operator
Re	Reynolds number
S	source term (coefficient)
$\mathbf{S}$	strain rate invariant
T	temperature
$\mathcal{T}$	time shift operator
u, v, w	velocities in x–, y–, z–direction respectively
U	upper part of matrix M
V	control cell / control volume
$\mathcal{V}$	volume of control cell V
W	generalized flow variable
$\vec{W}$	vector of conservative variables
x, y, z	cartesian coordinates of physical domain
Z	spatial discretization operator
$\mathcal{Z}$	Fourier symbol of spatial discretization operator Z

Greek Letters

α	angle of attack (without subscript)
α_*	Runge–Kutta stage coefficient (with subscript)
β, β^*	turbulence model parameters
γ	ratio of specific heats
δ	boundary layer thickness
Δt	time step
ΔW	difference of W from two consecutive time steps
Δx, Δy	grid spacing in x–, y–direction respectively
ε_*	coefficient for implicit residual smoothing
μ	viscosity coefficient / molecular viscosity
ξ, η	curvilinear coordinates of physical domain
ρ	density
σ, σ^*	turbulence model parameters
τ	time step per volume ($= \Delta t/V$)
τ_{wall}	shear stress on the wall
Φ_x, Φ_y	phase angles
ω	specific dissipation rate of turbulent kinetic energy k

Subscripts

∞	value at infinity / free stream value
c	convective (inviscid) term
ex	explicit treatment of source terms
FG	fine multigrid level
h, $2h$, ...	finest, resp. related coarser multigrid level
H, $2H$	fine but arbitrary, resp. related coarse multigrid level
i, j	nodal point indices
I, J	cell indices
IRS	implicit residual smoothing

l	laminar term
lim	limited turbulence value
MG	multigrid
$\vec{n}$	quantity in direction of $\vec{n}$
$\mathrm{pi-src}$	point implicit treatment of source terms
$\mathrm{pi-eq}$	point implicit treatment of whole equation
q	qth Runge–Kutta stage
ref	reference value
S	source term
SG	singlegrid
tu	turbulent term
v	viscous term
w	value at wall
x, y, z	cartesian coordinate directions

Superscripts

CG	coarse multigrid level
FG	fine multigrid level
h, $2h$, …	finest, resp. related coarser multigrid level
H, $2H$	fine but arbitrary, resp. related coarse multigrid level
MG	multigrid
n	value after nth time step
$\mathrm{pi-src}$	point implicit treatment of source terms
$\mathrm{pi-eq}$	point implicit treatment of whole equation
(q)	value after qth Runge–Kutta stage

Accents

$\bar{}$	(integral) mean value
$\hat{}$	amplitude of Fourier mode
$\vec{}$	vector
$\overline{=}$	tensor
$'$	(turbulent) fluctuation

Operators

$\bullet$	inner product of two tensors; scalar product of two vectors
$\circ$	application of an operator
∇	Nabla operator
div	divergence of a vector
$\big\vert_{*,*}$	at point $(*,*)$
$\big\vert^{*}$	in $*$–direction

1 Introduction

1.1 Problem Description

The complexity of numerical flow simulations in the aerodynamic context has been growing considerably over the last decades. This is mainly due to the growth of computational power available. Additionally further developed numerical algorithms and advanced physical models contributed. Nowadays flows around complex 3D aircraft configurations can be simulated on the basis of the time dependent Reynolds averaged Navier–Stokes (RANS) equations in combination with advanced turbulence models utilizing millions of grid points [68].

Complexity of aerodynamic simulation can be subdivided into geometrical complexity on the one hand and physical complexity on the other hand. Geometrical complexity is mainly related to detailed aircraft configurations including for instance deployed high lift devices and flap track fairings. Such geometrical details usually increase the complexity of physical phenomena to be simulated as for high lift flows [26, 106, 81]. Since grid generation becomes a crucial task for highly detailed aircraft configuration unstructured approaches are favorable.
The work at hand is focused on the physical complexity of aerodynamic flow simulation. This kind of complexity is not only influenced by the underlying geometry. Physical phenomena like dynamic stall or flows at flight Reynolds numbers contribute to this complexity just as the governing equations add to it. These equations describe physical models of fluid flow. In the eighties the robust and efficient numerical treatment of the Euler equations has been a challenge [56, 50, 63]. The next major step increasing the complexity of physical modeling has been the simulation of viscous flows governed by the Reynolds averaged Navier–Stokes (RANS) equations. The physical modeling may become even more complex in case of turbulent flows. Based on the Boussinesq hypothesis the RANS equations require the eddy viscosity to be a closed system of equations. The basic set of governing equations may be extended depending on the type of turbulence model in use. Algebraic turbulence models are based on empirical relations of local flow quantities. More advanced models use transport equations to provide the necessary information. In the latter case the RANS equations together with the turbulence transport equations form an extended set of governing equations. In the framework of numerical flow simulation one has to be aware that the combined application of RANS and turbulence equations on complex flow situations is often found not to be robust and efficient at the same time [42].

The simulation of viscous flows requires highly stretched grid cells, for instance in order to resolve efficiently gradients in boundary layers. Grid cells of this type are known

to reduce robustness and efficiency of numerical methods [12, 19], as (highly) stretched grid cells may introduce stiffness to the system of governing equations. When increasing the physical complexity of aerodynamic simulation by approaching flight Reynolds numbers an even thinner boundary layer has to be captured. Since the resolution of gradients in the boundary layer should be kept constant just as the number of grid points, such a reduced boundary layer thickness results in even more stretched grid cells than at wind tunnel Reynolds numbers. Therefore, a further reduction of robustness is to be expected.

Advanced turbulence models — i. e. transport equation turbulence models — are taken to address viscous effects more correctly. The turbulence equations contain source terms which in general are based on velocity gradients. Thus the quality of the resolution of gradients in boundary layers may significantly alter the resulting flow field. By an unsufficient resolution of velocity gradients, viscosity effects are not captured correctly in general. Moreover, the source terms of transport equation turbulence models — for instance the source terms of the k–ω turbulence model of Wilcox [131, 132] — scale with squared velocity gradients. This introduces an even larger dependence on the resolution of velocity gradients to the flow field.
Due to the source terms the turbulence equations are subject to a different characteristic time than the RANS equations [53]. In addition to stretched grid cells this increases the stiffness of the system of governing equations. Furthermore, turbulence source terms are active in the same regions of the flow where highly stretched grid cells deteriorate the properties of numerical methods.
These known effects combine to reduce dramatically the robustness of the numerical simulation of viscous flows around complex configurations utilizing RANS and turbulence equations.

While the numerical treatment of the RANS equations is said to have reached its limit concerning robustness and efficiency in the framework of industrial applications [42], the robust and efficient numerical solution of advanced turbulence model equations in the aerodynamic simulation process is still a research topic [27, 42, 53].

1.2 State of the Art

Experience has shown [105, 34] that the robust and efficient implementation of a turbulence model often requires slight modifications of the model itself, i. e. of its parameters or of the corresponding boundary conditions. This can be done correctly by turbulence modeling experts only. However, robustness problems are very often cured by a suitable adaptation of the numerical treatment of turbulence equations.

The numerical methods applied for complex aerodynamic simulations are mostly based on a finite volume approach in combination with the "method of lines". By the "method of lines" the spatial and the time derivatives occurring in the RANS and turbulence equations are treated separately. The spatial discretization of the governing equations results in a set of ordinary differential equations which has to be integrated in time. The corresponding time stepping schemes may be implicit or explicit and may use several kinds of speed–up techniques. Furthermore, neither the spatial discretization nor the time integration approach to the turbulence equations are necessarily the same

as for the RANS equations. Thus the whole variety of numerical methods applicable to the RANS equations may in principle be combined with the same variety of numerical methods applied to the turbulence equations. Correspondingly no comprehensive survey of the numerical treatment of turbulence equations has been found in the literature.
First investigations on modifying the spatial discretization of the turbulence equations and the scaling of the related artificial or numerical dissipation showed no encouraging effect. Thus the following investigations are focused on the time integration scheme and corresponding speed–up techniques applied to the turbulence equations.

Among the implicit time stepping schemes applied to turbulence equations, the ADI (alternating direction implicit) scheme is employed e. g. by flow solvers like CFL3D [60], TLNS3D [124, 121] and NPARC [133]. An implicit backward Euler time integration with a LU relaxation scheme for both the RANS and the turbulence equations is implemented in the elsA software [91].
Multistage schemes are typically used in explicit approaches [56] as incorporated e. g. in CANARI [24], EURANUS [27], ENSOLV [59] and FLOWer [62]. Multigrid — a standard convergence acceleration technique for (explicit) time integration schemes for the RANS equations [51, 115] — is also used to speed up time stepping schemes for turbulence equations. For explicit methods this is discussed e. g. in [73, 62, 27, 80, 38]. The combination of implicit time integration approaches with multigrid is presented for instance in [28, 41, 127]. Otherwise the turbulence equations are treated with singlegrid only like in [24, 27, 105, 91]. In this framework Couaillier et al. [27] applies a subiteration technique to the turbulence equations.
Many explicit time integration approaches incorporate special treatments of low implicitness for the turbulence equations. A point implicit treatment of the turbulence equations is applied in [80, 62, 87, 73]. Or the coefficients for implicit residual smoothing may be adapted to turbulence equations [27]. Other authors adapt the time step very accurately to the turbulence equations and rely on purely explicit approaches [69, 105].

The adaptation of multigrid transfer operators to others than the RANS equations is better known in the framework of chemically reacting flow simulations. Since chemistry model equations have a comparable structure to turbulence equations — both consist of convection, diffusion and source terms — their numerical treatment may show analogous problems related to numerical robustness.
In this context Edwards [32] applied a so–called "filtered" multigrid method which in particular separates modes depending on whether they are smooth or oscillating. The smooth ones are transferred to coarser grids while the oscillating modes are further damped on the current grid. Gerlinger et al. [40, 39] apply a local scaling factor while restricting the residuals needed for the forcing function for turbulence and chemistry equations. This scaling factor is partly based on source term values.

Further ideas to increase robustness of the numerical treatment of turbulence equations can be taken from literature: Especially to increase robustness of multigrid methods Mandal and Rajput [79] presented an approach which is intended to ensure stability of multigrid. In case several iterations are performed on coarser grids Mandal and Rajput model the influence of these coarse grid iterations on the forcing function. However, they applied this idea only to the Euler equations, and no extension to RANS equations has been given.

A completely different approach is proposed by Moulton and Steinhoff [89]: They modify the underlying physical model by introducing an additional vorticity conservation term. Based on this approach less stretched grid cells are needed to resolve a boundary layer. Thus the numerical properties of a time stepping scheme are less distorted by stretched grid cells resulting in higher efficiency and robustness. This method is called vorticity confinement, but has rather poor physical background.

For sure this is only a small summary of numerical treatments of turbulence equations used in some CFD codes. But, by this short survey it becomes obvious that no approach to integrate the turbulence equations in time is used more often than others. This surveyed variety suggests that the choice of the numerical treatment of turbulence equations depends more on personal experience than on generally acknowledged strategies. Therefore, a systematic investigation of numerical treatments of turbulence equations is still missing in literature. In addition, the simulation of flows at flight Reynolds numbers has not yet been reported.

1.3 Objectives

The aim of this work is to provide numerical algorithms that enable the robust and efficient simulation of turbulent flows around civil transport aircraft configurations at flight Reynolds numbers. In the framework of an industrially applied RANS solver the numerical treatment of advanced transport equation turbulence models is found to be the crucial task. This work intends to give more founded information about the background of robustness problems in the numerical treatment of turbulence equations. For this purpose a systematic investigation utilizing computations and Fourier analysis is performed. Based on the results of this study well suited numerical methods to treat the turbulence equations robustly and efficiently at the same time are derived.

The DLR flow solver FLOWer [62] is used as the basis for this work. It has been developed within the national German project "MEGAFLOW" [61, 66, 67, 68, 82, 83]. FLOWer is a block structured code able to solve the Reynolds averaged Navier–Stokes equations in combination with various turbulence models. The basic time integration scheme applied to both RANS and turbulence equations is an explicit 5–stage Runge–Kutta scheme sped up by local time stepping, multigrid and implicit residual smoothing. The FLOWer code has been intensively validated during the last years by DLR, universities and industry as documented e. g. in [61, 90, 3, 99, 88] for aircraft applications. Nowadays FLOWer is applied to 3D aircraft configurations as a daily routine in German aircraft industry [68, 3].
For the implementation of new features and other modifications in FLOWer as an industrially applied software some standards related to efficiency need to be satisfied: While increasing robustness, the efficiency of FLOWer in terms of CPU time and memory requirements is not to be reduced significantly. Additionally, the applicability of a new approach based on easy–to–use and case independent input parameters is preferred.

1.4 Overview

The compressible Navier–Stokes equations are introduced in Chap. 2. The changes they undergo due to Reynolds (and Favre) averaging are presented briefly. The governing equations are the compressible Reynolds averaged Navier–Stokes equations. This system is closed on the basis of the eddy viscosity hypothesis of Boussinesq [9]. The eddy viscosity is provided by some turbulence model. Therefore the system of governing equations is extended by some turbulence model equations if transport equation turbulence models are in use. The k–ω turbulence model of Wilcox [131, 132] is taken since it is a widely used representative of state–of–the–art turbulence models [21, 60, 27, 29, 42, 59, 74, 84, 105]. Its description is also given in Chap. 2.
The numerical treatment of the turbulence equations is discussed in Chap. 3 considering a scalar equation as an example. This scalar equation includes convective and diffusive terms in two dimensions besides a source term. The presentation of the computational methods applied to turbulence equations start at spatial discretization schemes for convection, diffusion and source terms. As a time stepping scheme an explicit Runge–Kutta method is employed. Speed–up techniques like implicit residual smoothing and multigrid are also included. The detailed presentation of all these methods is based on an operator notation. This operator notation is used in a consistent way in Chap. 4 to prepare a Fourier analysis of the numerical treatment of turbulence equations. First results from this analysis are discussed to give reasons for the choice of some basic settings for the Fourier analysis.
The identification of reasons for robustness problems in the simulation of turbulent flows when approaching flight Reynolds numbers is done in Chap. 5. These robustness problems are systematically investigated based on two dimensional computations and Fourier analyses and the related results are discussed. From this discussion several approaches to achieve a more robust numerical treatment of the turbulence equations are derived in Chap. 6. These approaches cover modifications to the multigrid method and to the time step as well. The coupling of the turbulence equations to the RANS equations is considered to be modified, while the explicit time stepping scheme is replaced by an implicit one. As implicit time stepping schemes a point implicit treatment of the whole turbulence equations is investigated apart from a line implicit and a fully implicit scheme. The latter is implemented based on a diagonal dominant alternating direction implicit (DDADI) method. All approaches are compared concerning their damping properties as well as their numerical behaviour.
In Chap. 7 the improved numerical treatment of the turbulence equations is applied to flight Reynolds number calculations for a 3D wing/body configuration. Additionally the gained robustness for complex cases of industrial relevance in general is demonstrated. Conclusions, Chap. 8, bring this work to its end.

2 Governing Equations

The numerical simulation of turbulent flows is based on the time dependent Navier–Stokes equations. These governing equations are derived from the fundamental conservation laws for mass, momentum and energy [46, 7].
For engineering applications it is unaffordable to treat the time dependent Navier–Stokes equations directly. The resolution in space and time necessary for this approach, called direct numerical simulation (DNS), still exceeds resources of modern high performance computing systems [7].
Thus a suitable approximation of the time dependent Navier–Stokes equations has to be taken: The Navier–Stokes equations are solved for mean values of the flow quantities following the approach of Reynolds [101]. He presented the first approach to treat turbulent flows approximately: The flow variables are decomposed into a mean and a fluctuating part, e. g. $\rho = \overline{\rho} + \rho'$ with an overbar denoting the mean value while the fluctuating part is marked by a prime. The mean values are obtained by a time averaging procedure called Reynolds averaging. As compressibility effects have to be considered for aerodynamic investigations of aircraft, the Reynolds averaging is used in combination with a mass averaging according to Favre (cf. [7], [105] or [115] and the references therein). The resulting so–called Reynolds averaged Navier–Stokes (RANS) equations are briefly presented in Sect. 2.1.

Due to the Reynolds averaging additional terms are introduced to the Navier–Stokes equations. Their determination is necessary to close the system of RANS equations. The eddy viscosity hypothesis of Boussinesq [9] results in a first–order closure of the system of governing equations. A suitable turbulence model is used to provide the eddy viscosity to the RANS equations.
In the past several kinds of first–order closure turbulence models have been devised, such as algebraic models e. g. by Baldwin and Lomax [4] or by Cebeci and Smith [22] or transport equation models. The transport equation turbulence models are categorized by their number of equations. The most widely used one equation turbulence model has been developed by Spalart and Allmaras [111]. It is based on an eddy viscosity like variable.
Two equation turbulence models in general are build from the transport equation for the turbulent kinetic energy and a suitable second transport equation. They are well established for CFD applications [74, 75, 80, 105]. In Sect. 2.2 the two equation k–ω turbulence model by Wilcox [131, 132] is introduced. It employs a specific dissipation rate as second quantity. This model has been chosen as a representative as it is a standard turbulence model for aircraft applications in a wide range of CFD codes [21, 60, 27, 29, 42, 59, 74, 84, 105]. However, the results of this work are not restricted to this turbulence model. A brief survey on turbulence models is provided by Hirsch

[47], while Blazek [7] introduces to the theoretical derivation of turbulence models.

While in Sect. 2.1 and in Sect. 2.2 the equations are given in integral form their differential formulation is given in Sect. 2.3. It will be used e. g. for Fourier analysis as presented in Chap. 4.
The chapter finishes by presenting a scalar example equation in Sect. 2.4 which is used in the subsequent chapters for the introduction to the numerical treatment of turbulence equations and the Fourier analysis.

2.1 Reynolds Averaged Navier–Stokes Equations

In this section the Reynolds averaged Navier–Stokes (RANS) equations and their main contributions are briefly discussed. Further insight on the physical background of the contributions to the flow equations is given by Hirsch [46] while Anderson [1] gives a detailed introduction on the relation between several physical points of view and the corresponding formulation of the Navier–Stokes equations.

In Sect. 2.1.1 the integral formulation of the Navier–Stokes equations is presented. This formulation constitutes the basis of the numerical method introduced in Chap. 3. The key advantage of the integral formulation is its validity also for flows with discontinuities like shocks and slip lines. Thus no assumptions on the continuity of the flow quantities need to be made [70]. The differences between the Navier–Stokes equations and their Reynolds averaged counterparts, the RANS equations, are described in Sect. 2.1.2.

2.1.1 Navier–Stokes Equations

The Navier–Stokes equations are a system of five equations for three dimensional flow (four equations for two dimensional flow). These equations describe the conservation of mass (one equation for the density ρ, the so–called continuity equation), the conservation of momentum (two or three equations — depending on the dimension — for the velocity) and the conservation of energy (one equation for the total energy E).

Following the derivation of conservation laws as given by Hirsch [46] the Navier–Stokes equations are obtained in an integral formulation based on the vector * of the conservative variables

$$\vec{W} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} \tag{2.1}$$

*Sometimes the term "tuple" is preferred since it is obvious that the solutions of the Navier–Stokes equations do not form a vector space. However, based on the theory of inertial manifolds [117] — which goes far beyond the scope of this work — solutions of the Navier–Stokes equations form a (nonlinear) manifold within a vector space. Thus the term "vector" is used here in accordance with nearly all other authors.

and read

$$\frac{d}{dt}\iiint_V \vec{W}\,\mathrm{d}V + \iint_{\partial V} \bar{\bar{F}} \bullet \vec{n}\,\mathrm{d}(\partial V) = 0 \tag{2.2}$$

for an arbitrary but fixed control volume V. As we are interested only in solutions at steady state within this work no time dependence of the control volume has to be taken into consideration. Thus we assume V to be constant in time and space. If this condition holds the time derivative in Eq. (2.2) can be put under the integral as

$$\iiint_V \frac{\partial}{\partial t}\vec{W}\,\mathrm{d}V + \iint_{\partial V} \bar{\bar{F}} \bullet \vec{n}\,\mathrm{d}(\partial V) = 0\,. \tag{2.3}$$

The surface of the control volume V is denoted by ∂V and the vector $\vec{n}$ is the outer normal unit vector of the surface element $d(\partial V)$.

The operator $\bullet$ is the inner product of two tensors. In the case of two tensors of first order it equals the scalar product of two vectors while for a second order and a first order tensor the inner product corresponds to a matrix vector multiplication [8]. Below it will become obvious that the continuity equation and the energy equation each contribute one tensor of first order to $\bar{\bar{F}}$ whereas the set of momentum equations provides a tensor of second order.

The tensor $\bar{\bar{F}}$ is a mathematical description of the net exchange (flux) of mass, momentum and energy across the boundary ∂V of the volume V. This flux tensor is given by

$$\bar{\bar{F}} = \bar{\bar{F}}_c - \bar{\bar{F}}_v \tag{2.4}$$

where $\bar{\bar{F}}_c$ and $\bar{\bar{F}}_v$ are the inviscid — usually called "convective" — and the viscous flux tensor, respectively. The inviscid flux tensor is defined by

$$\bar{\bar{F}}_c = \left(\vec{F}_{c,\,x}, \vec{F}_{c,\,y}, \vec{F}_{c,\,z}\right) \tag{2.5}$$

with the inviscid flux vectors

$$\vec{F}_{c,\,x} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho Hu \end{pmatrix}, \quad \vec{F}_{c,\,y} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho Hv \end{pmatrix} \quad \text{and} \quad \vec{F}_{c,\,z} = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho Hw \end{pmatrix}. \tag{2.6}$$

The total energy E is defined as the sum of the inner energy e and the kinetic energy of the flow:

$$E = e + \frac{u^2 + v^2 + w^2}{2} \tag{2.7}$$

with

$$e = c_v T \tag{2.8}$$

for a perfect gas. c_v denotes the specific heat at constant volume, and T is the temperature.

In Eq. (2.6) the total enthalpy

$$H = E + p/\rho \tag{2.9}$$

is also used. Assuming that air behaves as a calorically perfect gas the pressure p is given by the equation of state

$$p = (\gamma - 1)\rho \left(E - \frac{u^2 + v^2 + w^2}{2} \right) . \tag{2.10}$$

γ denotes the ratio of the specific heats c_p (at constant pressure) and c_v (at constant volume) and takes the value

$$\gamma = 1.4 \tag{2.11}$$

for air at standard conditions [110]. The temperature is given by

$$T = \frac{p}{\rho \, \mathtt{R}} \tag{2.12}$$

with the universal gas constant $\mathtt{R}$.

In analogy to the inviscid flux tensor $\bar{\bar{F}}_c$ (Eq. (2.5)) the viscous flux tensor $\bar{\bar{F}}_v$ is defined as

$$\bar{\bar{F}}_v = \left(\vec{F}_{v,x}, \vec{F}_{v,y}, \vec{F}_{v,z} \right) \tag{2.13}$$

with the viscous flux vectors

$$\vec{F}_{v,x} = \begin{pmatrix} 0 \\ \sigma_{xx} \\ \sigma_{yx} \\ \sigma_{zx} \\ \Psi_x \end{pmatrix} , \quad \vec{F}_{v,y} = \begin{pmatrix} 0 \\ \sigma_{xy} \\ \sigma_{yy} \\ \sigma_{zy} \\ \Psi_y \end{pmatrix} \quad \text{and} \quad \vec{F}_{v,z} = \begin{pmatrix} 0 \\ \sigma_{xz} \\ \sigma_{yz} \\ \sigma_{zz} \\ \Psi_z \end{pmatrix} . \tag{2.14}$$

For a Newtonian fluid the components σ_{**} of the viscous shear stress tensor are given as

$$\begin{aligned}
\sigma_{xx} &= 2\mu \frac{\partial u}{\partial x} - \frac{2}{3}\mu \, \mathrm{div}\,(\vec{q}) , \\
\sigma_{yy} &= 2\mu \frac{\partial v}{\partial y} - \frac{2}{3}\mu \, \mathrm{div}\,(\vec{q}) , \\
\sigma_{zz} &= 2\mu \frac{\partial w}{\partial z} - \frac{2}{3}\mu \, \mathrm{div}\,(\vec{q}) , \\
\sigma_{xy} &= \sigma_{yx} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) , \\
\sigma_{xz} &= \sigma_{zx} = \mu \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) , \\
\sigma_{yz} &= \sigma_{zy} = \mu \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) .
\end{aligned} \tag{2.15}$$

Here $\vec{q}$ is used as an abbreviation for the velocity vector

$$\vec{q} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} . \tag{2.16}$$

Furthermore, the divergence of the velocity is given by

$$\mathrm{div}\,(\vec{q}) = \nabla \bullet \vec{q} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} . \tag{2.17}$$

The viscosity μ is assumed to follow the Sutherland law

$$\mu = \mu_{\mathrm{ref}} \left(\frac{T}{T_{\mathrm{ref}}} \right)^{\frac{3}{2}} \frac{T_{\mathrm{ref}} + 110\,\mathtt{K}}{T + 110\,\mathtt{K}} \tag{2.18}$$

where μ_{ref} denotes the viscosity at the reference temperature T_{ref} [105]. The reference temperature is taken at the farfield boundary with a default value of $285\,\mathtt{K}$.

The viscous contributions to the energy equation are given as

$$\begin{aligned} \Psi_x &= u\sigma_{xx} + v\sigma_{xy} + w\sigma_{xz} + \mathtt{k}\frac{\partial T}{\partial x}\,, \\ \Psi_y &= u\sigma_{xy} + v\sigma_{yy} + w\sigma_{yz} + \mathtt{k}\frac{\partial T}{\partial y}\,, \\ \Psi_z &= u\sigma_{xz} + v\sigma_{yz} + w\sigma_{zz} + \mathtt{k}\frac{\partial T}{\partial z}\,. \end{aligned} \tag{2.19}$$

Here $\mathtt{k}$ denotes the coefficient of thermal conductivity given by

$$\mathtt{k} = c_p \frac{\mu}{\mathtt{Pr}} \tag{2.20}$$

with the Prandtl number $\mathtt{Pr} = 0.72$ for air.

Specification of boundary conditions can be found e. g. at [47, 1, 7] and the references therein.

2.1.2 Averaging of the Navier–Stokes Equations

In this section the Reynolds and Favre averaging of the Navier–Stokes equations is introduced very briefly. The resulting system of equations is not given as it is formally nearly equal to the system of Navier–Stokes equations but written for mean variables. The differences between the Navier–Stokes equations and their averaged counterpart, the RANS (Reynolds Averaged Navier–Stokes) equations, are presented in brevity.
For a detailed derivation and presentation of the averaging approaches applied to the Navier–Stokes equations resulting in the Reynolds averaged Navier–Stokes equations the reader is referred to the literature [7, 33, 105, 115] and the references therein.

Reynolds' approach to treat turbulent flows approximately [101] is based on a decomposition of the flow variables. The density for example is rewritten as

$$\rho = \overline{\rho} + \rho' \tag{2.21}$$

with an overbar denoting the mean value while the fluctuating part is marked by a prime. A time averaging of the flow quantities is applied to obtain the mean values,

$$\overline{\rho} = \lim_{\mathtt{T}\to\infty} \frac{1}{\mathtt{T}} \int_t^{t+\mathtt{T}} \rho\, dt\,. \tag{2.22}$$

Since compressibility effects have to be taken into account for aircraft applications a mass averaging according to Favre is applied to the velocity and the energy. By this Favre averaging

$$\tilde{u} = \frac{1}{\overline{\rho}} \lim_{\mathrm{T}\to\infty} \frac{1}{\mathrm{T}} \int_t^{t+\mathrm{T}} \rho u \, dt \tag{2.23}$$

the velocity u is decomposed as

$$u = \tilde{u} + u'' \,. \tag{2.24}$$

Here the tilde denotes the mean value while the double prime denotes the fluctuating part of the velocity. $\overline{\rho}$ — in the denominator of Eq. (2.23) — is the Reynolds averaged density given by Eq. (2.22).
One important relation between these averaging approaches is [7, 33]

$$\overline{\rho u} = \overline{\rho}\tilde{u} \,. \tag{2.25}$$

The general approach of averaging the Navier–Stokes equations is briefly demonstrated considering the total energy, Eq. (2.7), as an example:

1st step: Multiplying the total energy by ρ reads

$$\rho E = \rho e + \frac{1}{2}\left(\rho u^2 + \rho v^2 + \rho w^2\right) \tag{2.26}$$

2nd step: Averaging this equation according to Reynolds results in

$$\overline{\rho E} = \overline{\rho e} + \frac{1}{2}\left(\overline{\rho u^2} + \overline{\rho v^2} + \overline{\rho w^2}\right) \tag{2.27}$$

3rd step: Using Eq. (2.25) and canceling $\overline{\rho}$ leads to

$$\tilde{E} = \tilde{e} + \frac{1}{2}\left(\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2\right) + \tilde{k} \tag{2.28}$$

with the turbulent kinetic energy

$$\tilde{k} = \frac{1}{2}\left(\widetilde{[u'']^2} + \widetilde{[v'']^2} + \widetilde{[w'']^2}\right) \tag{2.29}$$

In order to ease the appearance of the so called Reynolds averaged Navier–Stokes equations neither $\bar{}$ nor $\tilde{}$ is used in the following. A detailed derivation of the RANS equations including all overbars and tildes is given e. g. in [33].

Additional terms are introduced to the Navier–Stokes equations by this averaging procedure: The viscous stress tensor, Eq. (2.15), is extended by the Reynolds stress tensor. It represents the transport of momentum due to turbulent fluctuations and is modeled based on the eddy viscosity hypothesis of Boussinesq [9]. This hypothesis states that the Reynolds stress tensor is linearly related to mean strain rate. The proportionality

factor is the eddy viscosity μ_{tu}. With this approach the viscous shear stress tensor is modified in its diagonal elements as

$$\begin{aligned}
\sigma_{xx} &= 2\mu\frac{\partial u}{\partial x} - \frac{2}{3}\mu\,\mathrm{div}\,(\vec{q}) - \frac{2}{3}\rho k\,, \\
\sigma_{yy} &= 2\mu\frac{\partial v}{\partial y} - \frac{2}{3}\mu\,\mathrm{div}\,(\vec{q}) - \frac{2}{3}\rho k\,, \\
\sigma_{zz} &= 2\mu\frac{\partial w}{\partial z} - \frac{2}{3}\mu\,\mathrm{div}\,(\vec{q}) - \frac{2}{3}\rho k\,.
\end{aligned} \tag{2.30}$$

Moreover, the viscosity μ equals the sum of the dynamic (also called laminar) viscosity μ_l and the eddy viscosity μ_{tu}:

$$\mu = \mu_l + \mu_{tu}\,. \tag{2.31}$$

While the eddy viscosity μ_{tu} is provided by means of a turbulence model, μ_l is assumed to follow the Sutherland law, Eq. (2.18).

The turbulent heat flux vector is added to the viscous energy flux vector and is modeled as

$$\mathtt{k}_{tu}\left(\frac{\partial T}{\partial x}\,,\,\frac{\partial T}{\partial y}\,,\frac{\partial T}{\partial z}\right)^{\mathsf{T}}\,, \tag{2.32}$$

where $\mathtt{k}_{tu}$ denotes the turbulent thermal conductivity coefficient given by

$$\mathtt{k}_{tu} = c_p\,\frac{\mu_{tu}}{\mathtt{Pr}_{tu}} \tag{2.33}$$

with the turbulent Prandtl number $\mathtt{Pr}_{tu} = 0.9$ for air. By extending Eq. (2.20) to

$$\mathtt{k} = \mathtt{k}_l + \mathtt{k}_{tu} = c_p\,\left(\frac{\mu_l}{\mathtt{Pr}} + \frac{\mu_{tu}}{\mathtt{Pr}_{tu}}\right) \tag{2.34}$$

the turbulent heat flux vector is included to the components of the viscous energy flux vector, Eq. (2.19), as

$$\begin{aligned}
\Psi_x &= u\sigma_{xx} + v\sigma_{xy} + w\sigma_{xz} + \mu_k\frac{\partial k}{\partial x} + \mathtt{k}\frac{\partial T}{\partial x}\,, \\
\Psi_y &= u\sigma_{xy} + v\sigma_{yy} + w\sigma_{yz} + \mu_k\frac{\partial k}{\partial y} + \mathtt{k}\frac{\partial T}{\partial y}\,, \\
\Psi_z &= u\sigma_{xz} + v\sigma_{yz} + w\sigma_{zz} + \mu_k\frac{\partial k}{\partial z} + \mathtt{k}\frac{\partial T}{\partial z}\,.
\end{aligned} \tag{2.35}$$

The contributions of the turbulent kinetic energy k — $\mu_k\partial k/\partial x$ etc. — in Eq. (2.35) describe the turbulent transport of k.

These extensions divide the RANS equations from the Navier–Stokes equations. In order to close the set of averaged governing equations only the eddy viscosity μ_{tu} is still necessary. A suitable turbulence model provides this eddy viscosity to the RANS equations. In the following section the k–ω turbulence model of Wilcox is introduced for this purpose.

2.2 Turbulence Model Equations

In the FLOWer code the k–ω turbulence model by Wilcox [131, 132] is implemented as the standard two equation eddy viscosity model. It was thoroughly investigated with respect to its performance and accuracy by Rudnik [105].

In order to add this transport equation turbulence model to the Navier–Stokes equations the vector $\vec{W}$ of the conservative variables is extended to seven variables as

$$\vec{W} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \\ \rho k \\ \rho \omega \end{pmatrix} . \tag{2.36}$$

The additional variables denote the turbulent kinetic energy k, Eq. (2.29), and the rate of dissipation of turbulence per unit energy ω. The integral formulation of the RANS equations in Eq. (2.3) is analogously extended to seven equations but with a source term:

$$\iiint\limits_{V} \frac{\partial}{\partial t}\vec{W}\,\mathrm{d}V + \iint\limits_{\partial V} \bar{\bar{F}} \bullet \vec{n}\,\mathrm{d}(\partial V) = \iiint\limits_{V} \vec{S}\left(\vec{W}\right)\,\mathrm{d}V\,. \tag{2.37}$$

$\vec{S}\left(\vec{W}\right)$ denotes a source term and is defined as

$$\vec{S}\left(\vec{W}\right) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ P_k + D_k \\ P_\omega + D_\omega \end{pmatrix} . \tag{2.38}$$

Thus the right hand side of the first five equations — the RANS equations — is still equal to zero. Only the turbulence equations contribute source terms which will be discussed below. The enlarged flux vectors (Eq. (2.6) and Eq. (2.14)) read

$$\vec{F}_{c,\,x} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho Hu \\ \rho ku \\ \rho \omega u \end{pmatrix}, \quad \vec{F}_{c,\,y} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho Hv \\ \rho kv \\ \rho \omega v \end{pmatrix} \quad \text{and} \quad \vec{F}_{c,\,z} = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho Hw \\ \rho kw \\ \rho \omega w \end{pmatrix} \tag{2.39}$$

for the inviscid fluxes. The viscous fluxes are now given as

$$\vec{F}_{v,x} = \begin{pmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{xz} \\ \Psi_x \\ \mu_k \dfrac{\partial k}{\partial x} \\ \mu_\omega \dfrac{\partial \omega}{\partial x} \end{pmatrix}, \quad \vec{F}_{v,y} = \begin{pmatrix} 0 \\ \sigma_{xy} \\ \sigma_{yy} \\ \sigma_{yz} \\ \Psi_y \\ \mu_k \dfrac{\partial k}{\partial y} \\ \mu_\omega \dfrac{\partial \omega}{\partial y} \end{pmatrix} \quad \text{and} \quad \vec{F}_{v,z} = \begin{pmatrix} 0 \\ \sigma_{xz} \\ \sigma_{yz} \\ \sigma_{zz} \\ \Psi_z \\ \mu_k \dfrac{\partial k}{\partial z} \\ \mu_\omega \dfrac{\partial \omega}{\partial z} \end{pmatrix} \tag{2.40}$$

with $\mu_k = \mu_l + \sigma^* \mu_{tu}$ and $\mu_\omega = \mu_l + \sigma \mu_{tu}$ while the eddy viscosity

$$\mu_{tu} = \frac{\rho k}{\omega} \tag{2.41}$$

is gained from the turbulence model. The parameters σ^* and σ are model dependent and are given at the end of this section together with other turbulence model parameters.

Each source term of the turbulence equations consists of a productive (P_*) and a destructive (D_*) part as denoted in Eq. (2.38). According to Wilcox the production of turbulent kinetic energy is driven by

$$P_k = \mu_{tu} \mathbf{S} - \frac{2}{3} \rho k \, \mathrm{div}\,(\vec{q}) \tag{2.42}$$

where $\mathbf{S}$ is the abbreviation of the strain rate invariant:

$$\begin{aligned} \mathbf{S} \;=\; & \frac{4}{3}\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2 - \frac{\partial u}{\partial x}\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\frac{\partial w}{\partial z} - \frac{\partial v}{\partial y}\frac{\partial w}{\partial z}\right] \\ & + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)^2 . \end{aligned} \tag{2.43}$$

The production of ω is given as

$$P_\omega = \frac{5}{9} \frac{\omega}{k} P_k \tag{2.44}$$

while its destruction is determined by

$$D_\omega = -\beta \rho \omega^2 . \tag{2.45}$$

The turbulent kinetic energy k is destructed under

$$D_k = -\beta^* \rho k \omega . \tag{2.46}$$

The turbulence model parameters have the empirical values according to Wilcox [131, 105, 34]:

$$\beta = 3/40, \;\; \beta^* = 9/100, \;\; \sigma = 1/2, \;\; \sigma^* = 1/2 . \tag{2.47}$$

Several boundary condition versions have been implemented to the FLOWer code. These boundary conditions and their specific differences are discussed in detail in [34].

At the inflow farfield k is computed from the free stream turbulence intensity Tu_∞ which can be specified by the user. As default $\mathsf{Tu}_\infty = 0.005$ is used when

$$k_\infty = \frac{3}{2}\left(\mathsf{Tu}_\infty\right)^2 \left(\vec{q}_\infty\right)^2 \tag{2.48}$$

is prescribed. The value for ω at the inflow farfield depends on k_∞ and reads at default

$$\omega_\infty = \frac{\rho_\infty k_\infty}{10^{-5}\mu_\infty}\,. \tag{2.49}$$

At solid walls the turbulent kinetic energy equals zero, $k_\mathrm{w} = 0$, while the value for ω_w is computed from a boundary condition suggested by Wilcox [131, 132]. This boundary condition is based on the dimensionless surface roughness. For more details the reader is referred to the literature for instance [34].
The outflow farfield boundary condition is defined by a 0th order extrapolation.

In the actual implementation to the FLOWer code the term $2/3\rho k$ has been neglected in the viscous shear stress tensor, Eq. (2.30). Different strategies to couple the k–ω turbulence model to the RANS equations will be discussed in Sect. 6.2.1.

2.3 Differential Formulation

The differential formulation of the governing equations is derived from the integral formulation Eq. (2.37) by means of Gauß' divergence theorem. A necessary assumption for the application of the divergence theorem is a continuous integrand. This condition is fulfilled if and only if no discontinuities such as shocks or contact discontinuities are present [1]. A discontinuous integrand can be treated in this context only by taking into account some additional conditions as presented briefly at the end of this section. Working from the assumption that the condition is fulfilled, i. e. the fluxes are continuous, Eq. (2.37) reads

$$\begin{aligned} & \iiint_V \frac{\partial}{\partial t}\vec{W}\,\mathrm{d}V + \iiint_V \mathrm{div}\left(\bar{\bar{F}}\right)\,\mathrm{d}V - \iiint_V \vec{S}\left(\vec{W}\right)\,\mathrm{d}V \\ = & \iiint_V \left(\frac{\partial}{\partial t}\vec{W} + \mathrm{div}\left(\bar{\bar{F}}\right) - \vec{S}\left(\vec{W}\right)\right)\,\mathrm{d}V \\ = & \; 0 \end{aligned} \tag{2.50}$$

after applying Gauß' divergence theorem to it.

As Eq. (2.50) should hold for arbitrary control volumes V it must be independent of the integration boundaries. This is only possible, if the integrand vanishes. Therefore it follows that

$$\frac{\partial}{\partial t}\vec{W} + \mathrm{div}\left(\bar{\bar{F}}\right) - \vec{S}\left(\vec{W}\right) = 0\,. \tag{2.51}$$

This equation can be rewritten as

$$\frac{\partial}{\partial t}\vec{W} + \frac{\partial \vec{F}_{c,x}}{\partial x} + \frac{\partial \vec{F}_{c,y}}{\partial y} + \frac{\partial \vec{F}_{c,z}}{\partial z} = \frac{\partial \vec{F}_{v,x}}{\partial x} + \frac{\partial \vec{F}_{v,y}}{\partial y} + \frac{\partial \vec{F}_{v,z}}{\partial z} + \vec{S}\left(\vec{W}\right)\,. \tag{2.52}$$

The restriction to flows without shocks or other discontinuities can be resolved if the Rankine–Hugoniot conditions are considered in addition to Eq. (2.52). They represent jump conditions at discontinuities [47]. Solving Eq. (2.52) coupled with the Rankine–Hugoniot equations as a local constraint at discontinuities yields the same range of applicability for the differential formulation of the governing equations as for their integral formulation.
A major drawback of these so–called "shock fitting" methods is to generate at run time a local area around discontinuities where the Rankine–Hugoniot conditions have to be applied.

2.4 Volume Averaged Scalar Model Equation

The whole set of Navier–Stokes and turbulence equations is solved in the FLOWer code by means of basically the same numerical method. However, the numerical treatment of the turbulence equations is the major topic of this work. As in the FLOWer code the k and the ω equation are treated in an uncoupled manner, the numerical method is introduced only for a scalar model equation for reasons of legibility. For the same reasons only two dimensions are considered. Most often the extensions to a system of equations or to three dimensions are straightforward. The differences in the numerical method for the different equations (RANS equations on the one hand and turbulence equations on the other hand) will be highlighted whenever necessary.

The equation to be used for the presentation of the numerical treatment of the turbulence equations is

$$\iint_V \frac{\partial}{\partial t} W \,\mathrm{d}V = -\int_{\partial V} \bar{\bar{F}} \bullet \vec{n} \,\mathrm{d}(\partial V) + \iint_V S\,(W)\ \mathrm{d}V \tag{2.53}$$

with the two dimensional flux tensor

$$\bar{\bar{F}} = \bar{\bar{F}}_c - \bar{\bar{F}}_v = (F_{c,x} - F_{v,x},\, F_{c,y} - F_{v,y}) = \left(uW - \mu \frac{\partial W}{\partial x},\, vW - \mu \frac{\partial W}{\partial y} \right) . \tag{2.54}$$

The scalar quantity W represents turbulence quantities as ρk or $\rho\omega$. u and v denote the x– resp. y–component of the convection velocity vector $\vec{q}$ in analogy to Eq. (2.16) whereas μ denotes the viscosity coefficient. Thus this flux tensor is a suitable representation of the turbulence components of the inviscid and viscous flux vectors given in Eqs. (2.39) and (2.40).

As the control cells V generally have arbitrary shape and therefore differing volumes, the numerical method has to be independent of the volume of a single control cell. This can be assured by assuming the time derivative of W to be volume averaged within each control cell, i. e. applying the first mean value theorem of integral calculus to the integral of the time derivative in Eq. (2.53). This yields

$$\iint_V \frac{\partial}{\partial t} W \,\mathrm{d}V = \overline{\frac{\partial}{\partial t} W} \cdot \iint_V \mathrm{d}V = \overline{\frac{\partial}{\partial t} W} \cdot \mathcal{V} \tag{2.55}$$

by which Eq. (2.53) is equivalent to

$$\overline{\frac{\partial}{\partial t}W} = -\frac{1}{\mathcal{V}}\int\limits_{\partial V}\bar{\bar{F}}\bullet\vec{n}\,\mathrm{d}(\partial V) + \frac{1}{\mathcal{V}}\iint\limits_{V} S\left(W\right)\,\mathrm{d}V\,. \tag{2.56}$$

$\overline{\partial W/\partial t}$ denotes the integral mean value of the time derivative of W over the cell V and $\mathcal{V}$ stands for the volume of V.

3 Numerical Treatment of Turbulence Equations

The Reynolds averaged Navier–Stokes equations introduced in Sect. 2.1, as well as the turbulence equations (cf. Sect. 2.2) are time dependent partial differential equations. The standard approach for solving time dependent partial differential equations is the so–called "method of lines" [2, 44, 56], i. e. the time discretization and the space discretization are treated uncoupled. For the method of lines the equations are spatially discretized in order to get a system of ordinary differential equations in time, which are continuous in time but discrete in space.

The physical space in which the solution of the RANS equations is to be calculated has to be discretized. This is done by distributing points over the physical domain. Connecting these points by straight lines generates a computational grid.
The FLOWer code can be applied to structured body–fitted multiblock grids generated by grid generation packages like e. g. MegaCads [17, 16].

In Sect. 3.1 the finite volume method, i. e. the spatial discretization of Eq. (2.37) is presented considering the scalar model equation, Eq. (2.56), as an example. The differences between the RANS and the turbulence equations in the discretization of the convective terms are pointed out. The time stepping scheme used for time integration of the semi–discretized equation is introduced in Sect. 3.2. The special treatment of the turbulence source terms is given here as well. The chapter is concluded by explaining the speed–up techniques applied to the fully discretized equation for enhancing robustness and efficiency in Sect. 3.3.

3.1 Finite Volume Method

In this section the spatial discretization of the turbulence equations are presented. A finite volume method is used to solve the equations. The idea of finite volume methods is to apply the integral formulation of the governing equations to each control volume on its own. Thus the conservation laws will be fulfilled in each cell.
In order to apply the finite volume method, the physical space in which the flowfield is to be calculated is covered by a set of control volumes defined by a computational grid. Presupposing non–overlapping control cells covering the computational domain without gaps it follows that an integral over the entire domain equals the sum over all integrals over each single cell.

As Eq. (2.56) is valid for arbitrary control volumes the volume averaged time derivative of W in a certain grid point (i,j) can be calculated by

$$\overline{\frac{\partial}{\partial t}W_{i,j}} = -\frac{1}{\widetilde{\mathcal{V}}_{i,j}} \int\limits_{\partial\widetilde{V}_{i,j}} \overline{\overline{F}} \bullet \vec{n}\, \mathrm{d}(\partial\widetilde{V}_{i,j}) + \frac{1}{\widetilde{\mathcal{V}}_{i,j}} \iint\limits_{\widetilde{V}_{i,j}} S\,(W)\ \mathrm{d}\widetilde{V}_{i,j}\,. \tag{3.1}$$

$\widetilde{V}_{i,j}$ has to be chosen as a suitable control cell around the grid node (i,j).

Various finite volume formulations are known in the literature [102, 65]. They distinguish from the arrangement of the control volumes as well as from the update points for the flow quantities. These schemes are the cell centered, the node centered and the cell vertex approach. For the cell centered approach the flow quantities are located at the center of the cell using the grid cell as the control volume. For both, the node centered and the cell vertex scheme, the flow variables are associated with the cell vertices but different control volumes are used. In general there are two possibilities to choose $\widetilde{V}_{i,j}$ within a node based framework: The first one is to take a supercell which is the union of all surrounding grid cells

$$\widetilde{V}_{i,j} = V_{I,J} \cup V_{I-1,J} \cup V_{I,J-1} \cup V_{I-1,J-1} \tag{3.2}$$

as shown in Fig. 3.1. This kind of control volume is employed for the central discretization of the inviscid terms of the RANS equations. The other possibility to define a suitable control cell is to use a compact cell

$$\widetilde{V}_{i,j} = V_{i,j} \tag{3.3}$$

which is sketched in Fig. 3.2. This cell is calculated by averaging cell face normal vectors. Hence its volume equals one fourth of the volume of the supercell given above. This definition of the compact cell is different from the one given in literature e. g. by Blazek [7]. Blazek defines the compact cell via its corner points. These are given as the cell centers of all cells surrounding grid point (i,j).

Fluxes across cell faces inside the computational domain cancel out each other when summing up the integrals over all control cells since the flux across an inner cell face contributes to the flux balances of the two adjacent control cells but with opposite signs. Thus both choices of control volumes — if correctly treated — lead to the same fluxes through the surface of the computational domain. As Hirsch [46] states only these fluxes affect the variations of the flow variables in the computational domain, not the flux values inside the domain. In addition to this boundary conditions must also be considered for these variations.
As the cell centered scheme has been newly implemented to the FLOWer code it is not introduced within this work. However, the cell centered approach and the compact cell approach are numerically very similar in the interior flow field. Thus the results derived within this work hold for cell centered schemes as well. This is demonstrated in Chap. 7.
The indices used in Eq. (3.2) resp. Eq. (3.3) belong to the grid cell — upper case indices (I,J) — or to the grid node — lower case indices (i,j). This notation is used throughout this work.

The spatial discretization operators for the right hand side of Eq. (3.1) will be derived below. An operator notation following Vichnevetsky and Bowles [128] is used in order

to get compact formulae and to achieve good readability. For this purpose a spatial index shift operator $\mathcal{E}\circ$ is introduced as

$$\mathcal{E}_i^r \circ W_{i,j} = W_{i+r,j} \quad \text{or} \quad \mathcal{E}_j^s \circ W_{i,j} = W_{i,j+s}\,. \tag{3.4}$$

In case r or s equals 1 the exponent is not written. Furthermore $\mathcal{I}\circ$ denotes the identity operator:

$$\mathcal{I} \circ W_{i,j} = W_{i,j}\,. \tag{3.5}$$

Like in Sect. 2.4 the scalar quantity W represents turbulence quantities as ρk or $\rho\omega$.
By means of these two operators the spatial discretization operator of the inviscid terms, $Z_c\circ$, is derived in Sect. 3.1.1. $Z_v\circ$, the operator of the spatial discretization of the viscous terms, is described in Sect. 3.1.2. In Sect. 3.1.3 it will be explained why the source terms are not spatially discretized and what this implies for $Z_S\circ$, the corresponding operator. Other operators will be introduced on demand.

One further remark needs to be given related to the implementation of the spatial discretization operators in the FLOWer code: The FLOWer code uses two ghost cells (also called dummy or halo cells) at any block boundary. By this approach the implementation of boundary conditions might become a little more tricky since not only the boundary points themselves but also the ghost points have to be set correctly in order to fulfill the boundary condition. However, any spatial discretization operator using not more than two neighboring points per direction may be implemented without degeneration of the discretization stencil at block boundaries: The same discretization operators are applied to all physical points in each block.

3.1.1 Spatial Discretization of Inviscid Terms

The spatial discretization of the inviscid terms is different for the turbulence equations and the Reynolds averaged Navier–Stokes equations. The latter one are subject to a second order central discretization for the inviscid fluxes according to Jameson et al. [56]. This approach is introduced briefly in the appendix, Sect. A.1.

Rudnik [105] compared two discretization schemes for transport equation turbulence models, a first order upwind scheme (AUSM, Advection Upstream Splitting Method [72, 64]) and a second order central discretization with artificial dissipation according to Jameson et al. [56]. He found that the second order central discretization does not always yield the expected higher spatial accuracy compared to the first order upwind scheme. Additionally the higher robustness of a first order discretization has been found to be more important than the higher spatial accuracy of a second order discretization [105, 107, 104].
The first order upwind scheme which is used to discretize the inviscid part of the turbulence equations is described in detail here. Although the AUSM scheme is implemented in the FLOWer code as well the default upwind discretization for turbulence equations is a simplified Roe–type scheme since it is computationally cheaper.

The surface integral of the inviscid flux, Eq. (2.53) resp. Eq. (2.54), is taken over the compact cell given in Eq. (3.3) which yields

$$\int\limits_{\partial\widetilde{V}_{i,j}} \bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}(\partial\widetilde{V}_{i,j}) = \int\limits_{\partial V_{i,j}} \bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}(\partial V_{i,j})\,. \tag{3.6}$$

This integral is split into four integrals over the cell faces of the control volume $V_{i,j}$ as sketched in Fig. 3.2:

$$\begin{aligned}\int\limits_{\partial V_{i,j}} \bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}(\partial V_{i,j}) &= \int\limits_{C_{i+\frac{1}{2},j}} \bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}C_{i+\frac{1}{2},j} - \int\limits_{C_{i-\frac{1}{2},j}} \bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}C_{i-\frac{1}{2},j} \\ &\quad + \int\limits_{C_{i,j+\frac{1}{2}}} \bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}C_{i,j+\frac{1}{2}} - \int\limits_{C_{i,j-\frac{1}{2}}} \bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}C_{i,j-\frac{1}{2}}\,.\end{aligned} \tag{3.7}$$

Here it is assumed that the normal unit vector $\vec{n}$ on a cell face is oriented in the positive direction of the corresponding curvilinear coordinate direction as shown in Fig. 3.3: The cell face normal vector e. g. on an $i = const$–face points towards the $i + 1 = const$–coordinate line.
By applying the mean value theorem of integral calculus to each of the surface integrals in Eq. (3.7) it follows e. g. for the cell face $C_{i+\frac{1}{2},j}$

$$\begin{aligned}\int\limits_{C_{i+\frac{1}{2},j}} \bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}C_{i+\frac{1}{2},j} &= \overline{\bar{\bar{F}}_{c;\,i+\frac{1}{2},j}} \bullet \vec{n}_{i+\frac{1}{2},j} \int\limits_{C_{i+\frac{1}{2},j}} \mathrm{d}C_{i+\frac{1}{2},j} \\ &= \overline{\bar{\bar{F}}_{c;\,i+\frac{1}{2},j}} \bullet \vec{n}_{i+\frac{1}{2},j}\,\mathcal{C}_{i+\frac{1}{2},j}\,.\end{aligned} \tag{3.8}$$

$\vec{n}_{i+\frac{1}{2},j}$ denotes the normal unit vector of the cell face $C_{i+\frac{1}{2},j}$ as given in Fig. 3.3. It is calculated as the mean value of the normal unit vectors of all $i = const$–faces surrounding the point $(i + \frac{1}{2}, j)$:

$$\vec{n}_{i+\frac{1}{2},j} = \frac{1}{4}\left(\vec{n}_{I+\frac{1}{2},J} + \vec{n}_{I+\frac{1}{2},J-1} + \vec{n}_{I-\frac{1}{2},J} + \vec{n}_{I-\frac{1}{2},J-1}\right)\,. \tag{3.9}$$

These vectors are shown in Fig. 3.4. In analogy to this the area of cell face $C_{i+\frac{1}{2},j}$ is defined:

$$\mathcal{C}_{i+\frac{1}{2},j} = \frac{1}{4}\left(\mathcal{C}_{I+\frac{1}{2},J} + \mathcal{C}_{I+\frac{1}{2},J-1} + \mathcal{C}_{I-\frac{1}{2},J} + \mathcal{C}_{I-\frac{1}{2},J-1}\right)\,. \tag{3.10}$$

Again Fig. 3.4 illustrates these cell faces. Using $_{i+\frac{1}{2},j}$ as an index for $\bar{\bar{F}}_c$ indicates that this value is taken on the cell face $C_{i+\frac{1}{2},j}$. As the normal unit vector $\vec{n}_{i+\frac{1}{2},j}$ and the cell face area $\mathcal{C}_{i+\frac{1}{2},j}$ are known at least the integral mean value of the flux tensor $\bar{\bar{F}}_c$ on the cell face $C_{i+\frac{1}{2},j}$ has to be approximated. Using an upwind scheme i. e. applying a discretization that for instance depends on the sign of the local convection velocity gives

$$\overline{\bar{\bar{F}}_{c;\,i+\frac{1}{2},j}} \approx \frac{1}{2}\left(\bar{\bar{F}}_{c;\,i+1,j} + \bar{\bar{F}}_{c;\,i,j}\right) - \frac{1}{2}\left|J_{c;\,i+\frac{1}{2},j}\right|\left(W_{i+1,j} - W_{i,j}\right) \tag{3.11}$$

with the Jacobian of the inviscid flux

$$J_c = \frac{\partial \bar{\bar{F}}_c}{\partial W}\,. \tag{3.12}$$

The notation of the upwind flux as given in Eq. (3.11) may be interpreted as the sum of a centrally discretized part — $1/2\left(\bar{\bar{F}}_{c;\,i+1,j}+\bar{\bar{F}}_{c;\,i,j}\right)$ — and some numerical dissipation* — $-1/2\left|J_{c;\,i+\frac{1}{2},j}\right|(W_{i+1,j}-W_{i,j})$ — [47].
Roe [7, 47, and references therein] introduced an approximative solver for the Riemann problem, also known as the shock tube problem. He decomposed the upwind flux across each cell face according to the Eigenvalues of the inviscid flux Jacobian J_c, i. e. he suggested to scale the numerical dissipation with these Eigenvalues: Assuming a transformation matrix P to exist with $J = P\mathcal{J}P^{-1}$, where $\mathcal{J}$ denotes the diagonal matrix of eigenvalues of J, the absolute value of the Jacobian in Eq. (3.11) is defined as

$$|J| = P\,|\mathcal{J}|\,P^{-1}\,. \tag{3.13}$$

As the inviscid flux of the scalar model equation (compare Eq. (2.54)) is defined by

$$\bar{\bar{F}}_c = W\vec{q}^{\,\mathsf{T}} \tag{3.14}$$

the Jacobian J equals $\vec{q}^{\,\mathsf{T}}$. Its approximation on the cell face $C_{i+\frac{1}{2},j}$ by an arithmetic average of the velocity,

$$J_{c;\,i+\frac{1}{2},j} \approx \vec{q}^{\,\mathsf{T}}_{i+\frac{1}{2},j} = \frac{\vec{q}^{\,\mathsf{T}}_{i+1,j}+\vec{q}^{\,\mathsf{T}}_{i,j}}{2}\,, \tag{3.15}$$

suggests itself. It may happen that this averaged velocity goes to zero, for instance in case of

$$\vec{q}^{\,\mathsf{T}}_{i+1,j} \to -\vec{q}^{\,\mathsf{T}}_{i,j}\,. \tag{3.16}$$

This would result in a central discretization of the inviscid flux without dissipation which is known to be unstable [63, 56]. One possibility to cure this problem is based on some minimum value for the scaling factor of the numerical dissipation [64, 96].
Another one is used here: Instead of taking the exact left and right state — $\bar{\bar{F}}_{c;\,i,j}$ and $\bar{\bar{F}}_{c;\,i+1,j}$ respectively — for the local shock tube problem based on the flow variable, W, and the velocity, $\vec{q}^{\,\mathsf{T}}$,

$$\bar{\bar{F}}_{c;\,i,j} = W_{i,j}\vec{q}^{\,\mathsf{T}}_{i,j} \qquad \text{and} \qquad \bar{\bar{F}}_{c;\,i+1,j} = W_{i+1,j}\vec{q}^{\,\mathsf{T}}_{i+1,j} \tag{3.17}$$

they are approximated as

$$\bar{\bar{F}}_{c;\,i,j} \approx W_{i,j}\vec{q}^{\,\mathsf{T}}_{i+\frac{1}{2},j} \qquad \text{and} \qquad \bar{\bar{F}}_{c;\,i+1,j} \approx W_{i+1,j}\vec{q}^{\,\mathsf{T}}_{i+\frac{1}{2},j}\,. \tag{3.18}$$

This approximation corresponds to a locally constant convection velocity $\vec{q}_{i+\frac{1}{2},j}$ between (i,j) and $(i+1,j)$. However, this leads to an approximation of the integral mean value of the flux tensor $\bar{\bar{F}}_c$ on the cell face $C_{i+\frac{1}{2},j}$ by

$$\overline{\bar{\bar{F}}_{c;\,i+\frac{1}{2},j}} \approx \frac{1}{2}\vec{q}^{\,\mathsf{T}}_{i+\frac{1}{2},j}\,(W_{i+1,j}+W_{i,j}) - \frac{1}{2}|\vec{q}|^{\,\mathsf{T}}_{i+\frac{1}{2},j}\,(W_{i+1,j}-W_{i,j}) \tag{3.19}$$

instead of Eq. (3.11). This simplified upwind scheme ensures stability whether the scaling factor of the numerical dissipation tends to zero or not since the central part is

*One should distinguish between artificial dissipation as introduced by Jameson et al. [56] for the stabilization of a central discretization scheme and numerical dissipation which is inherent to upwind discretization schemes but can be seen only if the upwind scheme is written as in Eq. (3.11).

scaled in the same way.
Thus the upwind scheme used within the FLOWer code for the convection terms of the turbulence equations is a simplified Roe scheme. The absolute value of the velocity vector in Eq. (3.19) is taken for each component separately in order to assure upwinding in each coordinate direction.

Using the operator notation introduced in Sect. 3.1 Eq. (3.19) reads

$$\overline{\overline{\overline{F}}}_{c;\,i+\frac{1}{2},j} \approx \frac{1}{2}\vec{q}^{\,\mathsf{T}}_{i+\frac{1}{2},j}\left(\mathcal{E}_i\circ\,+\mathcal{I}\circ\right)W_{i,j} - \frac{1}{2}|\vec{q}|^{\,\mathsf{T}}_{i+\frac{1}{2},j}\left(\mathcal{E}_i\circ\,-\mathcal{I}\circ\right)W_{i,j}\,. \tag{3.20}$$

The velocity normal to a cell face in the direction of the cell face normal unit vector is defined as

$$q_{\vec{n}} = \vec{q}^{\,\mathsf{T}} \bullet \vec{n} = (u,v) \bullet \begin{pmatrix} n_x \\ n_y \end{pmatrix} = un_x + vn_y\,. \tag{3.21}$$

From Eq. (3.20) together with Eq. (3.8) and Eq. (3.21) the discretization operator $Z_c|^i\circ$ of the i–direction of Eq. (3.7) is given as

$$\begin{aligned} Z_c|^i\circ \;\;=\;\; & \frac{1}{2}\mathcal{C}_{i+\frac{1}{2},j}\left[q_{\vec{n};\,i+\frac{1}{2},j}\left(\mathcal{E}_i\circ\,+\mathcal{I}\circ\right) - |q|_{\vec{n};\,i+\frac{1}{2},j}\left(\mathcal{E}_i\circ\,-\mathcal{I}\circ\right)\right] \\ & - \frac{1}{2}\mathcal{C}_{i-\frac{1}{2},j}\left[q_{\vec{n};\,i-\frac{1}{2},j}\left(\mathcal{I}\circ\,+\mathcal{E}_i^{-1}\circ\right) - |q|_{\vec{n};\,i-\frac{1}{2},j}\left(\mathcal{I}\circ\,-\mathcal{E}_i^{-1}\circ\right)\right]\,. \end{aligned} \tag{3.22}$$

Keep in mind that the absolute value of $\vec{q}$ — which is the approximated Jacobian of the inviscid flux (cmp. Eq. (3.15)) — has to be taken componentwise. From there it follows

$$|q|_{\vec{n}} = |u|\cdot n_x + |v|\cdot n_y\,. \tag{3.23}$$

With the discretization operator $Z_c|^j\circ$ of the j–direction,

$$\begin{aligned} Z_c|^j\circ \;\;=\;\; & \frac{1}{2}\mathcal{C}_{i,j+\frac{1}{2}}\left[q_{\vec{n};\,i,j+\frac{1}{2}}\left(\mathcal{E}_j\circ\,+\mathcal{I}\circ\right) - |q|_{\vec{n};\,i,j+\frac{1}{2}}\left(\mathcal{E}_j\circ\,-\mathcal{I}\circ\right)\right] \\ & - \frac{1}{2}\mathcal{C}_{i,j-\frac{1}{2}}\left[q_{\vec{n};\,i,j-\frac{1}{2}}\left(\mathcal{I}\circ\,+\mathcal{E}_j^{-1}\circ\right) - |q|_{\vec{n};\,i,j-\frac{1}{2}}\left(\mathcal{I}\circ\,-\mathcal{E}_j^{-1}\circ\right)\right]\,, \end{aligned} \tag{3.24}$$

the approximation of Eq. (3.7) can be abbreviated as

$$\int\limits_{\partial V_{i,j}} \overline{\overline{F}}_c \bullet \vec{n}\,\mathrm{d}(\partial V_{i,j}) \approx Z_c\circ W_{i,j} = \left[\,Z_c|^i\circ\,+\,Z_c|^j\circ\,\right]W_{i,j}\,. \tag{3.25}$$

Since the surface integral over the convective flux tensor $\overline{\overline{F}}_c$ has a negative sign in Eq. (3.1) the operators $Z_c|^i\circ$ and $Z_c|^j\circ$ are redefined in order to hide signs within the operator notation of the spatial discretizations: The operator of the discretization of the convective terms in i–direction then reads

$$\begin{aligned} Z_c|^i\circ \;\;=\;\; & \frac{1}{2}\mathcal{C}_{i-\frac{1}{2},j}\left[q_{\vec{n};\,i-\frac{1}{2},j}\left(\mathcal{I}\circ\,+\mathcal{E}_i^{-1}\circ\right) - |q|_{\vec{n};\,i-\frac{1}{2},j}\left(\mathcal{I}\circ\,-\mathcal{E}_i^{-1}\circ\right)\right] \\ & - \frac{1}{2}\mathcal{C}_{i+\frac{1}{2},j}\left[q_{\vec{n};\,i+\frac{1}{2},j}\left(\mathcal{E}_i\circ\,+\mathcal{I}\circ\right) - |q|_{\vec{n};\,i+\frac{1}{2},j}\left(\mathcal{E}_i\circ\,-\mathcal{I}\circ\right)\right]\,. \end{aligned} \tag{3.26}$$

Its analogue in j–direction is now given as

$$\begin{aligned} Z_c|^j\circ &= \frac{1}{2}\mathcal{C}_{i,j-\frac{1}{2}}\left[q_{\vec{n};\,i,j-\frac{1}{2}}\left(\mathcal{I}\circ + \mathcal{E}_j^{-1}\circ\right) - |q|_{\vec{n};\,i,j-\frac{1}{2}}\left(\mathcal{I}\circ - \mathcal{E}_j^{-1}\circ\right)\right] \\ &\quad - \frac{1}{2}\mathcal{C}_{i,j+\frac{1}{2}}\left[q_{\vec{n};\,i,j+\frac{1}{2}}\left(\mathcal{E}_j\circ + \mathcal{I}\circ\right) - |q|_{\vec{n};\,i,j+\frac{1}{2}}\left(\mathcal{E}_j\circ - \mathcal{I}\circ\right)\right] \end{aligned} \tag{3.27}$$

thus they approximate

$$-\int\limits_{\partial V_{i,j}} \overline{\overline{F}}_c \bullet \vec{n}\,\mathrm{d}(\partial V_{i,j}) \approx Z_c\circ W_{i,j} = \left[Z_c|^i\circ + Z_c|^j\circ\right] W_{i,j}\,. \tag{3.28}$$

3.1.2 Spatial Discretization of Viscous Terms

The viscous flux as given in Eq. (2.40) is spatially discretized in the same way for all governing equations. Its surface integral is taken over the compact cell given in Eq. (3.3) like the convective flux of the turbulence equations, which yields

$$\int\limits_{\partial \widetilde{V}_{i,j}} \overline{\overline{F}}_v \bullet \vec{n}\,\mathrm{d}(\partial \widetilde{V}_{i,j}) = \int\limits_{\partial V_{i,j}} \overline{\overline{F}}_v \bullet \vec{n}\,\mathrm{d}(\partial V_{i,j})\,. \tag{3.29}$$

Therefore this integral is transformed in the same way as the inviscid surface integral in Eq. (3.6) with the difference that the viscous fluxes are subject to a thin layer approximation: Within a thin layer approximation only gradients transverse to the mean flow direction are taken into account presupposing other shear stress terms to have a negligible effect on the flow behaviour. Assuming j to be the coordinate direction normal to the viscous shear layers only the surface integrals over $C_{i,j+\frac{1}{2}}$ and $C_{i,j-\frac{1}{2}}$ need to be considered. Thus the transformations result in

$$\int\limits_{\partial V_{i,j}} \overline{\overline{F}}_v \bullet \vec{n}\,\mathrm{d}(\partial V_{i,j}) = \int\limits_{C_{i,j+\frac{1}{2}}} \overline{\overline{F}}_v \bullet \vec{n}\,\mathrm{d}C_{i,j+\frac{1}{2}} - \int\limits_{C_{i,j-\frac{1}{2}}} \overline{\overline{F}}_v \bullet \vec{n}\,\mathrm{d}C_{i,j-\frac{1}{2}} \tag{3.30}$$

and further

$$\begin{aligned} \int\limits_{C_{i,j+\frac{1}{2}}} \overline{\overline{F}}_v \bullet \vec{n}\,\mathrm{d}C_{i,j+\frac{1}{2}} &= \overline{\overline{F}_{v;\,i,j+\frac{1}{2}}} \bullet \vec{n}_{i,j+\frac{1}{2}} \int\limits_{C_{i,j+\frac{1}{2}}} \mathrm{d}C_{i,j+\frac{1}{2}} \\ &= \overline{\overline{F}_{v;\,i,j+\frac{1}{2}}} \bullet \vec{n}_{i,j+\frac{1}{2}}\,\mathcal{C}_{i,j+\frac{1}{2}} \end{aligned} \tag{3.31}$$

which is a viscous analogue to Eq. (3.8). Still an approximation of $\overline{\overline{F}_{v;\,i,j+\frac{1}{2}}}$ is needed. Remember the definition of $\overline{\overline{F}}_v$ for the scalar model equation (compare Eq. (2.54)) is

$$\overline{\overline{F}}_v = \mu\left(\frac{\partial W}{\partial x}, \frac{\partial W}{\partial y}\right)\,. \tag{3.32}$$

Although the viscous flux tensor for the RANS equations differs from Eq. (3.32) the following considerations in general hold for it, too.

μ is approximated on the cell face $C_{i,j+\frac{1}{2}}$ by the mean value

$$\mu_{i,j+\frac{1}{2}} \approx \frac{\mu_{i,j+1} + \mu_{i,j}}{2} \,. \tag{3.33}$$

The derivatives of W on the cell face $C_{i,j+\frac{1}{2}}$ with respect to x and y are computed using a local transformation from cartesian coordinates (x, y) to generalized curvilinear coordinates (ξ, η) as given by Radespiel et al. [97]. This coordinate transformation results in

$$\left(\frac{\partial W}{\partial x}, \frac{\partial W}{\partial y}\right)^{\mathsf{T}}_{i,j+\frac{1}{2}} \approx (W_{i,j+1} - W_{i,j}) \, \frac{\frac{1}{4} \sum\limits_{l=I-1}^{I} \left[(\mathcal{C}\vec{n})_{l,J+\frac{1}{2}} + (\mathcal{C}\vec{n})_{l,J-\frac{1}{2}} \right]}{\mathcal{V}_{i,j+\frac{1}{2}}} \,. \tag{3.34}$$

The details of the derivation of Eq. (3.34) are found in the appendix, Sect. A.2. $\mathcal{C}\vec{n}$ is the product of the cell face area $\mathcal{C}$ and the cell face normal unit vector $\vec{n}$. This product defines a cell face according to its spatial alignment and area. $\mathcal{V}_{i,j+\frac{1}{2}}$ denotes the volume of the auxiliary control cell $V_{i,j+\frac{1}{2}}$ as sketched in Fig. 3.2. Thus

$$\mathcal{V}_{i,j+\frac{1}{2}} = \frac{1}{2} \left(\mathcal{V}_{I,J} + \mathcal{V}_{I-1,J} \right) \,. \tag{3.35}$$

Further we define

$$(\mathcal{C}\vec{n})^{*}_{i,j+\frac{1}{2}} = \frac{\frac{1}{4} \sum\limits_{l=I-1}^{I} \left[(\mathcal{C}\vec{n})_{l,J+\frac{1}{2}} + (\mathcal{C}\vec{n})_{l,J-\frac{1}{2}} \right]}{\mathcal{V}_{i,j+\frac{1}{2}}} \,. \tag{3.36}$$

Now the approximation of the integral mean value of the viscous flux $\overline{\overline{F}}_v$ can be agglomerated from Eq. (3.33), Eq. (3.34) and Eq. (3.36) as

$$\overline{\overline{F}}_{v;\,i,j+\frac{1}{2}} \approx \mu_{i,j+\frac{1}{2}} \, (\mathcal{C}\vec{n})^{*}_{i,j+\frac{1}{2}} \, (W_{i,j+1} - W_{i,j}) \,. \tag{3.37}$$

In operator notation this formula reads

$$\overline{\overline{F}}_{v;\,i,j+\frac{1}{2}} \approx \mu_{i,j+\frac{1}{2}} \, (\mathcal{C}\vec{n})^{*}_{i,j+\frac{1}{2}} \left(\mathcal{E}_j \circ \, - \mathcal{I} \circ \right) W_{i,j} \,. \tag{3.38}$$

Inserting this formula into Eq. (3.31) yields

$$\int\limits_{C_{i,j+\frac{1}{2}}} \overline{\overline{F}}_v \bullet \vec{n} \, \mathrm{d}C_{i,j+\frac{1}{2}} \approx \mu_{i,j+\frac{1}{2}} \, (\mathcal{C}\vec{n})^{*}_{i,j+\frac{1}{2}} \bullet \vec{n}_{i,j+\frac{1}{2}} \, \mathcal{C}_{i,j+\frac{1}{2}} \left(\mathcal{E}_j \circ \, - \mathcal{I} \circ \right) W_{i,j} \,. \tag{3.39}$$

Here the definitions for $\vec{n}_{i,j+\frac{1}{2}}$ and for $\mathcal{C}_{i,j+\frac{1}{2}}$ as given in Sect. 3.1.1 (Eq. (3.9) resp. Eq. (3.10)) are to be considered.
Analogously an operator notation for the discretized viscous flux across $\mathcal{C}_{i,j-\frac{1}{2}}$ can be derived as

$$\int\limits_{C_{i,j-\frac{1}{2}}} \overline{\overline{F}}_v \bullet \vec{n} \, \mathrm{d}C_{i,j-\frac{1}{2}} \approx \mu_{i,j-\frac{1}{2}} \, (\mathcal{C}\vec{n})^{*}_{i,j-\frac{1}{2}} \bullet \vec{n}_{i,j-\frac{1}{2}} \, \mathcal{C}_{i,j-\frac{1}{2}} \left(\mathcal{I} \circ \, - \mathcal{E}_j^{-1} \circ \right) W_{i,j} \,. \tag{3.40}$$

Inserting the last two equations into Eq. (3.30) yields the corresponding discretization operator $Z_v|^j\circ$ of the viscous flux in j–direction to be

$$\begin{aligned} Z_v|^j\circ \quad = \quad & \mu_{i,j+\frac{1}{2}}\,(\mathcal{C}\vec{n})^*_{i,j+\frac{1}{2}} \bullet \vec{n}_{i,j+\frac{1}{2}}\,\mathcal{C}_{i,j+\frac{1}{2}}\left(\mathcal{E}_j\circ\, - \mathcal{I}\circ\right) \\ & - \mu_{i,j-\frac{1}{2}}\,(\mathcal{C}\vec{n})^*_{i,j-\frac{1}{2}} \bullet \vec{n}_{i,j-\frac{1}{2}}\,\mathcal{C}_{i,j-\frac{1}{2}}\left(\mathcal{I}\circ\, - \mathcal{E}_j^{-1}\circ\right) . \end{aligned} \tag{3.41}$$

As Eq. (3.29) is subject to the thin layer approximation its discretization reads in operator notation

$$\int\limits_{\partial V_{i,j}} \bar{\bar{F}}_v \bullet \vec{n}\,\mathrm{d}(\partial V_{i,j}) \approx Z_v\circ W_{i,j} = Z_v|^j\circ W_{i,j}\,. \tag{3.42}$$

3.1.3 Spatial Discretization of Source Terms

The volume integral of the source term in Eq. (3.1)

$$\iint\limits_{\widetilde{V}_{i,j}} S\,(W)\;\mathrm{d}\widetilde{V}_{i,j} \tag{3.43}$$

is transformed by means of the first mean value theorem of integral calculus like the time derivative in Eq. (2.55). This results in

$$\iint\limits_{\widetilde{V}_{i,j}} S\,(W)\;\mathrm{d}\widetilde{V}_{i,j} = \overline{S\,(W_{i,j})}\cdot \iint\limits_{\widetilde{V}_{i,j}} \mathrm{d}\widetilde{V}_{i,j} = \overline{S\,(W_{i,j})}\cdot \widetilde{\mathcal{V}}_{i,j}\,. \tag{3.44}$$

As the source terms of the turbulence equations represent volume sources they are strictly local quantities only related to a grid point and its surrounding control volume. In other words the source term at one grid point does not depend on source terms at any other (neighboring) grid point. In this sense source terms are not to be spatially discretized.
However, turbulence source terms usually depend on gradients of flow quantities. For the calculation of gradients of course neighboring grid points are necessary. Variations of this kind of spatial discretization of source terms may yield improvements as discussed e. g. in [125]. One has to keep in mind that the shallow water equations treated in [125] do not include other gradients than those needed for the source terms.
In the framework of viscous flows gradients are needed not only for turbulence source terms, but also for viscous fluxes of both RANS and turbulence equations. Hence a modification of this kind of spatial discretization of the source terms is not discussed in this work in order to keep the computation of gradients consistent.

As noted turbulence source terms are volume sources. Thus they are assumed to be constant within the corresponding control cell $\widetilde{V}_{i,j}$. This leads to

$$\overline{S\,(W_{i,j})} \approx S\,(W_{i,j})\;. \tag{3.45}$$

The source terms given in Eqs. (2.42) – (2.45) are nonlinear functions of the turbulence quantities. Thus it is not possible to rewrite the approximated version of Eq. (3.44)

in (linear) operator notation as introduced in Sect. 3.1. The corresponding nonlinear operator is

$$Z_S \circ \; = \widetilde{\mathcal{V}}\Big|_{*,*} S\,(\mathcal{I} \circ\,) \tag{3.46}$$

with S being a nonlinear function. However, in case of a linear source term the corresponding discretization operator $Z_S \circ$ is a linear operator:

$$Z_S \circ \; = \widetilde{\mathcal{V}}\Big|_{*,*} S\mathcal{I} \circ \tag{3.47}$$

with S denoting the source term coefficient. Since the Fourier analysis as introduced in Chap. 4 is able to deal with linear problems only, Eq. (3.47) whenever results from this analysis are presented.
Note that the control cell $\widetilde{V}\Big|_{*,*}$ has not been specified as the semi–discretization of the source terms does not depend on it. The indices are chosen according to the indices of the point, to which the discretization operator $Z_S \circ$ of the source term is applied to.

The source terms contain spatial derivatives of the velocities which are computed in the same way as the derivatives within the viscous fluxes, Eq. (3.34).

3.2 Time Stepping Scheme

The residual of a time–independent partial differential equation

$$L(u) = b \tag{3.48}$$

is defined by

$$R = L(\widetilde{u}) - b \tag{3.49}$$

with L denoting the differential operator and $\widetilde{u}$ being some (numerical) approximation to the analytical solution u.
The same definition of the residual would hold for the governing equations at steady state ($b = \partial W/\partial t = 0$). As in this work emphasis is laid on the numerical treatment of the governing equations, the residual is defined as the spatially discretized right hand side of Eq. (3.1):

$$R_{i,j} = (Z_c \circ \; + Z_v \circ \; + Z_S \circ\,)\, W_{i,j} = Z \circ W_{i,j}\,. \tag{3.50}$$

$Z_* \circ$ are the spatial discretization operators of the convective, the diffusive and the source terms respectively. They are given by Eq. (3.28), Eq. (3.42) and Eq. (3.46). By $Z \circ$ the sum of all spatial discretization operators is denoted. Here and in all following sections the overbar denoting integral mean values is skipped in order to ensure readability.

Although we are interested only in steady state solutions of the governing equations[†], i. e. solutions which fulfill $\partial W/\partial t = 0$, the time derivative is not neglected in Eq. (2.56). Instead of this the time dependency is used to apply a time marching method in order to drive the residual towards zero since the residual as defined in Eq. (3.50) denotes

[†]Questions concerning existence and uniqueness of the steady state solutions of the system of RANS and turbulence equations go far beyond the scope of this work. Thus both existence and uniqueness of these steady state solutions are taken for granted.

the change of the solution during one iteration step related to the time step, i. e. the residual R approximates the time derivative $\partial W/\partial t$:

$$\frac{\partial W}{\partial t} \approx \frac{1}{\widetilde{\mathcal{V}}} R = \frac{1}{\widetilde{\mathcal{V}}} Z \circ W \,. \tag{3.51}$$

All equations, i. e. the RANS as well as the turbulence equations, are integrated in time by the same time stepping scheme. In the FLOWer code an explicit hybrid five stage Runge–Kutta scheme [63] with point implicit treatment of the turbulence source terms has been implemented.

In the following sections the operator $\mathcal{G}_{\mathrm{ex}}^{(q=5)} \circ$ describes the Runge–Kutta time stepping scheme (Sect. 3.2.1) while $\mathcal{G}_{\mathrm{pi-src}}^{(q=5)} \circ$ includes the point implicit treatment of the source terms (Sect. 3.2.2). Both are introduced as polynomials of the spatial discretization operators $Z_c \circ$, $Z_v \circ$ and $Z_S \circ$ and of the temporal shift operator $\mathcal{T} \circ$.

The time step used within the Runge–Kutta scheme is described in Sect. 3.3.1.

3.2.1 Runge–Kutta Time Stepping Scheme

The explicit time discretization of the semi–discretization of Eq. (2.56) as it is used in the FLOWer code is given by

$$\frac{W_{i,j}^{n+1} - W_{i,j}^{n}}{\Delta t_{i,j}} = \frac{1}{\widetilde{\mathcal{V}}_{i,j}} \left(Z_c \circ + Z_v \circ + Z_S \circ \right) W_{i,j}^{n} \,. \tag{3.52}$$

As noted above the lower indices i and j are denoting a grid point. The newly introduced superscript n denotes a value after the nth time step. The analogue holds for $n+1$.

$$\tau = \frac{\Delta t}{\widetilde{\mathcal{V}}} \tag{3.53}$$

is used as an abbreviation for the time step per control volume. An explicit hybrid five stage Runge–Kutta scheme [56, 63] with evaluation of the viscous term and the source term on the first stage only — since both terms require the calculation of gradients, which is very expensive in terms of CPU time — is applied to Eq. (3.52):

$$\begin{aligned}
W^{(0)} &= W^{n} \\
W^{(1)} &= W^{(0)} + \alpha_1 \tau \Big[Z_c \circ + Z_v \circ + Z_S \circ \Big] W^{(0)} \\
W^{(2)} &= W^{(0)} + \alpha_2 \tau \Big[Z_c \circ W^{(1)} + \left(Z_v \circ + Z_S \circ \right) W^{(0)} \Big] \\
W^{(3)} &= W^{(0)} + \alpha_3 \tau \Big[Z_c \circ W^{(2)} + \left(Z_v \circ + Z_S \circ \right) W^{(0)} \Big] \\
W^{(4)} &= W^{(0)} + \alpha_4 \tau \Big[Z_c \circ W^{(3)} + \left(Z_v \circ + Z_S \circ \right) W^{(0)} \Big] \\
W^{(5)} &= W^{(0)} + \alpha_5 \tau \Big[Z_c \circ W^{(4)} + \left(Z_v \circ + Z_S \circ \right) W^{(0)} \Big] \\
W^{n+1} &= W^{(5)} \,.
\end{aligned} \tag{3.54}$$

The α_q are the so–called stage coefficients of the Runge–Kutta scheme. They can be optimized with respect to smoothing properties [71] or to time accuracy [44]. The nodal

point indices i and j are skipped as the time stepping scheme is independent of the grid point. An abbreviated form of the qth stage

$$W^{(q)} = W^{(0)} + \alpha_q \tau R^{(q-1)} \tag{3.55}$$

is used with the corresponding qth stage residual

$$R^{(q-1)} = Z_c \circ W^{(q-1)} + (Z_v \circ + Z_S \circ) W^{(0)} . \tag{3.56}$$

By successive insertion as carried out in detail in Sect. A.3 the qth stage of the Runge–Kutta scheme Eq. (3.54) can be cast into its characterizing operator polynomial

$$f_{\mathrm{ex}}^{(q)} \circ = \alpha_q \mathcal{I} \circ + \alpha_q \alpha_{q-1} \tau Z_c \circ + \ldots + \prod_{r=1}^{q} \alpha_r \, (\tau Z_c \circ)^{q-1} . \tag{3.57}$$

The subscript $_{\mathrm{ex}}$ of $f \circ$ is used to indicate the explicit character of the Runge–Kutta stage. $^{(q)}$ denotes the stage number within the Runge–Kutta scheme. Now the qth stage is abbreviated as

$$W^{(q)} = W^{(0)} + f_{\mathrm{ex}}^{(q)} \circ \tau R^{(0)} . \tag{3.58}$$

Thus Eq. (3.54) can be rewritten in operator notation as

$$W^{n+1} = \mathcal{G}_{\mathrm{ex}}^{(q=5)} \circ W^{n} \tag{3.59}$$

with

$$\mathcal{G}_{\mathrm{ex}}^{(q=5)} \circ = \mathcal{I} \circ + f_{\mathrm{ex}}^{(q=5)} \circ \tau Z \circ . \tag{3.60}$$

This operator notation of a $(q = 5)$stage Runge–Kutta scheme takes into account neither the point implicit treatment of the source term nor the implicit residual smoothing. The latter will be introduced in Sect. 3.3.2 whereas the point implicit scheme is discussed in the following section.

3.2.2 Point Implicit Time Discretization of Source Terms

The basic idea of a point implicit treatment of a source term is introduced for a simple model problem in Sect. 3.2.2.1. The turbulence source terms are treated implicitly in the FLOWer code in order to overcome severe time step restrictions they introduce to explicit time stepping schemes.

The source terms $P_k + D_k$ and $P_\omega + D_\omega$ (Eqs. (2.42) – (2.45)) are treated point implicitly, independent of all other terms as shown in Sect. 3.2.2.2. This treatment differs from the method proposed by Mavriplis and Martinelli [80] who apply a point implicit approach to all terms of the turbulence equations. Their approach is investigated together with other more implicit methods for the turbulence equations in Sect. 6.3.

The implementation of the source term derivative matrix in the FLOWer code presented in Sect. 3.2.2.3 corresponds to the work of Merci et al. [87] and to Wilcox [132] because the approximated source term Jacobian contains only contributions of negative, i. e. destructive source term parts.

3.2.2.1 Basic idea of Point Implicit Treatment of Source Terms

As no spatial dependency among turbulence source terms is given any implicit treatment of these source terms reduces to a point implicit one. The point implicit treatment yields a quasi–implicit scheme since a transformation to an explicit expression for the new time level is easily possible.

The basic idea is explained with the following 1D convection equation including a source term:

$$\frac{\partial W}{\partial t} = -u\frac{\partial W}{\partial x} + S(W)\,.$$

A first order upwind scheme yields for $\Delta W = W_i^{n+1} - W_i^n$ with the source term $S(W)$ treated implicitly:

$$\begin{aligned}
\frac{\Delta W}{\Delta t} &= -u_i \frac{1}{\Delta x}\left(W_i^n - W_{i-1}^n\right) + S\left(W_i^{n+1}\right) \\
&\approx -u_i \frac{1}{\Delta x}\left(W_i^n - W_{i-1}^n\right) + S\left(W_i^n\right) + \Delta W \frac{\partial S}{\partial W}\left(W_i^n\right) \qquad (3.61)\\
\Rightarrow\ \Delta W &= \underbrace{\left(\frac{1}{\Delta t}\cdot \bar{\bar{I}} - \frac{\partial S}{\partial W}\left(W_i^n\right)\right)^{-1}}_{\Delta t_{\mathrm{pi}}} \left[-u_i \frac{1}{\Delta x}\left(W_i^n - W_{i-1}^n\right) + S\left(W_i^n\right)\right]\,. \qquad (3.62)
\end{aligned}$$

The linearization of the source term around W_i^n (compare Eq. (3.61)) can be interpreted as a change of the local time step for the turbulence equations ($\Delta t \rightarrow \Delta t_{\mathrm{pi}}$) as shown in Eq. (3.62). As the approximation to the source term Jacobian $\partial S/\partial W$ contains negative terms only $\Delta t_{\mathrm{pi}} \leq \Delta t$ holds. $\bar{\bar{I}}$ denotes the identity matrix. The source term Jacobian $\partial S/\partial W$ is presented in Sect. 3.2.2.3 for the k–ω turbulence model.

3.2.2.2 Runge–Kutta Scheme with Point Implicit Source Term Treatment

The application of a multistage scheme like described in Sect. 3.2.1 to a turbulence equation with a point implicitly treated source term is not strictly straightforward. Treating source terms point implicitly, as introduced in the previous section, changes Eq. (3.52) to

$$W^{n+1} = W^n + \tau\left(Z_c\circ + Z_v\circ + Z_S\circ \mathcal{T}\circ\right) W^n \qquad (3.63)$$

with the time shift operator

$$\mathcal{T}\circ W^n = W^{n+1}\,. \qquad (3.64)$$

Again the nodal point indices have been dropped and τ as defined in Eq. (3.53) has been used.
Now one has to decide whether to linearize $Z_S\circ\mathcal{T}\circ W^n = Z_S\circ W^{n+1}$ around W^n before applying the Runge–Kutta scheme or to apply first the multistage scheme and to linearize on the qth stage $Z_S\circ W^{(q)}$ with respect to $W^{(0)}$ (= W^n) afterwards. The latter case is implemented in the FLOWer code and is discussed below. The linearization of the point implicitly treated source term before the Runge–Kutta scheme is applied will be discussed in Sect. 6.2.3.
Theoretically a linearization of $Z_S\circ W^{(q)}$ with respect to $W^{(q-1)}$ is also possible. This

would result in a recalculation of the source terms on each Runge–Kutta stage, what was already ruled out in Sect. 3.2.1.

The qth stage reads

$$W^{(q)} = W^{(0)} + \alpha_q \tau \left(Z_c \circ W^{(q-1)} + Z_v \circ W^{(0)} + Z_S \circ W^{(q)} \right) \tag{3.65}$$

before linearizing $Z_S \circ W^{(q)}$. After the linearization one has

$$W^{(q)} = W^{(0)} + \alpha_q^{\mathrm{pi-src}} \tau R^{(q-1)} \tag{3.66}$$

with

$$\alpha_q^{\mathrm{pi-src}} = \frac{\alpha_q}{1 - \alpha_q \tau \dfrac{\partial \left(Z_S \circ W^{(0)} \right)}{\partial W}} \,. \tag{3.67}$$

$R^{(q-1)}$ is taken from Eq. (3.56). $\partial (Z_S \circ W) / \partial W$ in Eq. (3.67) corresponds to the source term derivative with respect to W, i. e. it is related mainly to the derivative of the nonlinear source term function S.

In analogy to Eq. (3.57) the operator polynomial characterizing the qth stage as given in Eq. (3.66) reads

$$f^{(q)}_{\mathrm{pi-src}} \circ \; = \alpha_q^{\mathrm{pi-src}} + \alpha_q^{\mathrm{pi-src}} \alpha_{q-1}^{\mathrm{pi-src}} \tau Z_c \circ \; + \ldots + \prod_{r=1}^{q} \alpha_r^{\mathrm{pi-src}} \left(\tau Z_c \circ \right)^{q-1} \,. \tag{3.68}$$

The derivation of this polynomial is briefly described in Sect. A.3.1. The subscript $_{\mathrm{pi-src}}$ of $f\circ$ denotes the point implicit treatment of the source term. The time stepping scheme operator now reads

$$\mathcal{G}^{(q=5)}_{\mathrm{pi-src}} \circ \; = \mathcal{I} \circ \; + f^{(q=5)}_{\mathrm{pi-src}} \circ \tau Z \circ \,. \tag{3.69}$$

3.2.2.3 Approximation of Source Term Jacobian

The main problem in the framework of the point implicit treatment of the turbulence source terms concerns the linearization of the implicit term $S(W_i^{n+1})$. From a calculational point of view the approximation of the source term Jacobian is most relevant. In the FLOWer code the approach of Mavriplis and Martinelli [80] has been implemented, i. e. the eddy viscosity μ_{tu} has been taken to be constant for the linearization. Under this assumption the source term Jacobian for the k–ω turbulence model reads [105]

$$\begin{pmatrix} \dfrac{\partial S_k}{\partial(\rho k)} & \dfrac{\partial S_k}{\partial(\rho \omega)} \\ \dfrac{\partial S_\omega}{\partial(\rho k)} & \dfrac{\partial S_\omega}{\partial(\rho \omega)} \end{pmatrix} = \begin{pmatrix} -\dfrac{2}{3} \operatorname{div}(\vec{q}) - \beta^* \omega & -\beta^* k \\ 0 & -\dfrac{5}{9}\dfrac{2}{3} \operatorname{div}(\vec{q}) - 2\beta\omega \end{pmatrix} \tag{3.70}$$

with $\operatorname{div}(\vec{q})$ defined as in Eq. (2.17).
In order to increase the diagonal dominance of this matrix the entry on the upper diagonal is set to zero, i. e.

$$\frac{\partial S_k}{\partial \rho \omega} = 0 \,.$$

Additionally $\operatorname{div}(\vec{q})$ is only taken into account if its value is positive.
An analogous but slightly different approximation of the source term Jacobian is given by Wilcox [132].

3.2.3 Limiting of Turbulence Variables

As the turbulence variables represent physical quantities which are non–negative by definition, this property has to be kept throughout the whole simulation process. Additionally the eddy viscosity calculated from these quantities always exists in real situations, i. e. is a non–zero positive finite value. From these assumptions one deduces that both turbulence quantities — in addition to the physical requirement of being non–negative — have to be kept positive at any time.
Due to the source terms inherited from the turbulence model equations, especially due to the destructive part of the source terms, the positivity of the turbulence variables k and ω can not be guaranteed all the time during the iteration process. In order to ensure this necessary property, a lower bound is introduced after any update of the turbulence quantities. Therefore after each Runge–Kutta stage q the positivity of k and ω is enforced by

$$k_{\text{lim}}^{(q)} = \max\left(k^{(q)}, \text{lim}_k \cdot k_\infty\right) \tag{3.71}$$

for the turbulent kinetic energy k and by

$$\omega_{\text{lim}}^{(q)} = \max\left(\omega^{(q)}, \text{lim}_\omega \cdot \omega_\infty\right) \tag{3.72}$$

for the specific dissipation rate ω. lim_* are input parameters to the FLOWer code with default values equal to 10^{-5}. With the subscript $_\infty$ the corresponding free stream values of k and ω are denoted.

3.3 Speed–Up Techniques

Within this work only results at steady state are sought. Thus no time accuracy is needed during the convergence process. As all speed–up techniques presented here destroy time accuracy they can only be applied to the time integration of steady problems. Or they are used for speeding up the pseudo time stepping inside of the dual time stepping approach for time accurate computations [52]. In Sect. 3.3.1 the local time stepping is introduced while implicit residual smoothing is explained in Sect. 3.3.2. The multigrid method is described in Sect. 3.3.3.

3.3.1 Local Time Stepping

Local time stepping means to use the maximum time step allowed by local stability requirements in each control volume. Therefore the time stepping scheme operates at its stability limit everywhere in the flowfield [29]. Time accuracy is lost but a higher convergence speed is achieved as information is transported through each cell within one time step. Otherwise (global time stepping) the time step has to fulfill the stability requirements of the smallest grid cell.
Since the time steps used in the FLOWer code are calculated as

$$\Delta t_{i,j} = \frac{\widetilde{\mathcal{V}}_{i,j}^2}{\widetilde{\mathcal{V}}_{i,j} \cdot (\lambda_\xi + \lambda_\eta) + \texttt{DTVI} \cdot \dfrac{\mu_{l;i,j} + \mu_{tu;i,j}}{\rho_{i,j}} \left(\mathcal{C}_\xi \cdot \mathcal{C}_\xi + \mathcal{C}_\eta \cdot \mathcal{C}_\eta\right)} \tag{3.73}$$

both convection and diffusion stability limits are included [94]. Here the characteristic speed

$$\lambda_* = |\vec{q} \bullet (\mathcal{C}\vec{n})_*| + a_{i,j} \cdot \mathcal{C}_* \tag{3.74}$$

is calculated using averages over the super cell defined in Eq. (3.2) of the velocity and the cell faces where

$$a_{i,j} = \sqrt{\frac{1}{\rho_{i,j}} p_{i,j} \gamma} \tag{3.75}$$

defines the local speed of sound.
A more detailed derivation of the calculated time step is given in [94, 97, 105].

`DTVI` is an input parameter of FLOWer and weights the influence of the parabolic eigenvalue, i. e. the influence of the diffusion stability limit on the time step.
Another FLOWer input parameter is the value of the Courant number `CFL` which is limited by the CFL (Courant–Friedrichs–Levy) condition. The upper limit of `CFL` depends mainly on the time stepping scheme, but can be extended by the application of implicit residual smoothing as introduced in the following section. The definition of the time step per volume in Eq. (3.53) is changed by inclusion of the CFL number:

$$\tau = \texttt{CFL}\, \frac{\Delta t}{\widetilde{\mathcal{V}}} \,. \tag{3.76}$$

3.3.2 Implicit Residual Smoothing

The implicit residual smoothing within the FLOWer code is based on [54] by Jameson et al. and works with variable or constant smoothing coefficients. In two dimensions this smoothing reads

$$-\varepsilon_i R^{\dagger}_{i-1,j} + (1+2\varepsilon_i) R^{\dagger}_{i,j} - \varepsilon_i R^{\dagger}_{i+1,j} = R^{(q-1)}_{i,j} \tag{3.77a}$$

$$-\varepsilon_j R^{\ddagger}_{i,j-1} + (1+2\varepsilon_j) R^{\ddagger}_{i,j} - \varepsilon_j R^{\ddagger}_{i,j+1} = R^{\dagger}_{i,j} \tag{3.77b}$$

with $R^{(q-1)}_{i,j}$ denoting the qth stage residual in node (i,j) defined in Eq. (3.56). ε_* is a constant or variable smoothing coefficient. "variable" means that the smoothing coefficient may change from grid point to grid point i. e. $\varepsilon_* = \varepsilon_{*;\,i,j}$. There are several possibilities to define suitable variable smoothing coefficients but they are not discussed here. In this regard the reader is referred to the literature [95, 120, 115, 97, 98, 29].

Eq. (3.77a) is to be resolved for $R^{\dagger}_{i,j}$ whereas Eq. (3.77b) is to be resolved for $R^{\ddagger}_{i,j}$:

$$R^{\dagger}_{i,j} = B_i^{-1} \circ R^{(q-1)}_{i,j} \tag{3.78a}$$

$$R^{\ddagger}_{i,j} = B_j^{-1} \circ R^{\dagger}_{i,j} = B_j^{-1} \circ B_i^{-1} \circ R^{(q-1)}_{i,j} \tag{3.78b}$$

$B_i\circ$ and $B_j\circ$ stand for the implicit smoothing operators in i– and j–direction respectively. In operator notation they read

$$B_i\circ = -\varepsilon_i \mathcal{E}_i^{-1}\circ + (1+2\varepsilon_i)\mathcal{I}\circ - \varepsilon_i \mathcal{E}_i \circ \tag{3.79}$$

respectively

$$B_j\circ = -\varepsilon_j \mathcal{E}_j^{-1}\circ + (1+2\varepsilon_j)\mathcal{I}\circ - \varepsilon_j \mathcal{E}_j \circ \,. \tag{3.80}$$

As the coordinate directions are treated one after the other the whole implicit residual smoothing is reached by multiplying the coordinate dependent operators:

$$B\circ \; = B_i\circ \, B_j\circ \; . \tag{3.81}$$

The implicit residual smoothing operator depends on the smoothing coefficients

$$B\circ \; = B(\varepsilon_i, \varepsilon_j)\circ \tag{3.82}$$

although this is not explicitly mentioned for better legibility.

The Runge–Kutta schemes described in Sect. 3.2 do not take into account implicit smoothing of the residuals. This can be introduced by applying the smoothing to the stage residual on each stage in Eq. (3.54). Because of this the characterizing operator polynomial of the qth stage of the Runge–Kutta scheme with point implicit source term treatment changes to

$$\begin{aligned} f^{(q)}_{\mathrm{pi-src,\,IRS}}\circ \quad &= \quad B^{-1}\circ \left[\alpha_q^{\mathrm{pi-src}} + \alpha_q^{\mathrm{pi-src}}\alpha_{q-1}^{\mathrm{pi-src}}\tau Z_c\circ B^{-1}\circ \right. \\ &\qquad \left. + \ldots + \prod_{r=1}^{q} \alpha_r^{\mathrm{pi-src}} \left(\tau Z_c\circ B^{-1}\circ\right)^{q-1} \right] . \end{aligned} \tag{3.83}$$

The derivation of this polynomial is briefly described in Sect. A.3.2. If the implicit residual smoothing is not applied on every Runge–Kutta stage as it is possible in the FLOWer code via user specification, the resulting operator differs from Eq. (3.83). Since the application of this smoothing on every stage has been the default of the FLOWer code other possible settings have not been investigated.

By use of this modified polynomial the operator of the Runge–Kutta scheme with point implicit source term treatment and implicit residual smoothing can be written as

$$\mathcal{G}^{(q=5)}_{\mathrm{pi-src,\,IRS}}\circ \; = \mathcal{I}\circ \; + f^{(q=5)}_{\mathrm{pi-src,\,IRS}}\circ \, \tau Z\circ \; . \tag{3.84}$$

3.3.3 Multigrid Method

In the FLOWer code the multigrid method is applied not only to the RANS equations but also to the turbulence equations by default. This multigrid method is introduced in detail in this section. It is presented in an operator notation based on the operators derived in the previous sections.
Since many introductions to multigrid exist e. g. [10, 14, 57, 114, 118, 130], the reader is referred to the literature for deeper insight into the potential of multigrid methods.

Multigrid methods are built from two major ingredients: One is a so–called smoother. The other ingredient is at least one computational grid coarser than the baseline computational grid.
A smoother is a numerical method which is able to smooth a quantity, i. e. the smoother is able to reduce the amplitudes of high–frequency components in particular of any quantity the smoother is applied to. For instance iterative methods like Gauss–Seidel or time stepping schemes like the Runge–Kutta scheme introduced in Sect. 3.2 are well

known to have good smoothing properties [129, 55, 71, 112, 115, 114, 118]. In general the quantity the smoother is applied to is the error $e^n = W^n - W^\infty$.
The second ingredient to multigrid methods are grids coarser than the baseline computational grid: Smooth quantities do not necessarily need the fine resolution of the baseline grid. They can also be well approximated on a coarser grid. However, the smooth error may appear unsmooth on the coarser grid. Applying the smoother to the error on the coarser grid results in reduced amplitudes of high–frequency error components related to the coarse grid. These high–frequency error components are different from the high–frequency error components on the baseline grid [14, 118, 130].
Additionally numerical methods are (substantially) less expensive on coarse grids since these grids consist of (substantially) less grid points compared to the baseline grid.

In the FLOWer code the smoother is a Runge–Kutta time stepping scheme (Sect. 3.2). Within the framework of structured grids as used in the FLOWer code a coarse grid is usually generated by skipping every second grid line in each curvilinear coordinate direction. This approach is called standard coarsening [118].

In the FLOWer code the nonlinear version of the multigrid method, the full approximation scheme (FAS), is implemented. A two grid cycle consists of

- (pre–) smoothing on the fine grid
  - Runge–Kutta step(s) with implicit residual smoothing
- restriction of data from the fine to the coarse grid
  - computation of residuals on the fine grid for the forcing function
  - injection of variables including turbulence source terms and eddy viscosity
  - volume weighted restriction of residuals used for constructing the forcing function
- smoothing on the coarse grid
  - modify right hand side by the forcing function
  - Runge–Kutta step(s) with implicit residual smoothing
- prolongation of data from the coarse to the fine grid
  - computation of the coarse grid correction
  - linear interpolation of the coarse grid correction
- (post–) smoothing on the fine grid
  - smoothing of the prolongated coarse grid corrections
  - update of the variables by the smoothed coarse grid corrections

Since the coarse grid corrections might be unsmooth on the fine grid because of the linear interpolation used for the transfer from the coarse to the fine grid, an implicit smoothing step (as introduced in the previous section) with constant coefficients is applied to these transferred values before updating the variables. In order to treat the complete multigrid correction, i. e. the coarse grid correction as well as the correction calculated by the Runge–Kutta scheme on the fine grid, in the same way, this implicit smoothing is also applied to the Runge–Kutta correction.

In the following sections all operators not introduced by now but necessary for the description of a two level multigrid cycle are presented.

3.3.3.1 Fine Grid Operator

In contrast to the implicit smoothing operator multigrid does not change the characterizing operator polynomial $f^{(q)}\circ$ of the Runge–Kutta scheme, Eq. (3.54). The multigrid method modifies the operator $\mathcal{G}_{\mathrm{ex}}^{(q=5)}\circ$ of the Runge–Kutta scheme, Eq. (3.60) resp. Eq. (3.84), which equals a single grid operator.

From Eq. (3.84) it is known that the variable value after one Runge–Kutta step on one grid level is given by

$$W_{h,\,\mathrm{SG}}^{n+1} = W_h^n + f_{\mathrm{pi-src},\,\mathrm{IRS},\,h}^{(q=5)} \circ \tau_h Z_h \circ W_h^n \,. \tag{3.85}$$

The supplementary subscript h indicates fine grid values whereas the operator $Z\circ$ denotes the sum of the spatial discretization operators as already introduced in Eq. (3.50):

$$Z\circ \;= Z_c\circ \;+ Z_v\circ \;+ Z_S\circ \;. \tag{3.86}$$

The fine grid residuals are implicitly smoothed with constant coefficients before being transferred to the coarse grid. Therefore the variable $W_{h,\,\mathrm{SG}}^{n+1}$ changes to

$$W_{h,\,\mathrm{FG}}^{n+1} = W_h^n + B_{\mathrm{MG}}^{-1}\circ f_{\mathrm{pi-src},\,\mathrm{IRS},\,h}^{(q=5)} \circ \tau_h Z_h \circ W_h^n \tag{3.87}$$

which is the variable value after one Runge–Kutta cycle only on the finest grid within a multigrid cycle. This is denoted by the subscript $_{\mathrm{FG}}$. Here $B_{\mathrm{MG}}\circ$ is the same operator as $B_h\circ$ but depends on the constant coefficients $(\varepsilon_{\mathrm{MG}}, \varepsilon_{\mathrm{MG}})$ instead of the variable coefficients $(\varepsilon_i, \varepsilon_j)$. Radespiel [94] noted that $\varepsilon_{\mathrm{MG}} = 0.2$ is a reasonable value.

The operator $\mathcal{G}^{(q=5)}\circ$ of the time stepping scheme on the fine grid within a multigrid cycle — which is here denoted by the superscript $^{\mathrm{FG}}$ — reads

$$\mathcal{G}_{\mathrm{pi-src},\,\mathrm{IRS},\,h}^{(q=5),\,\mathrm{FG}}\circ \;= \mathcal{I}\circ \;+ B_{\mathrm{MG}}^{-1}\circ f_{\mathrm{pi-src},\,\mathrm{IRS},\,h}^{(q=5)} \circ \tau_h Z_h\circ \;. \tag{3.88}$$

Thus Eq. (3.87) reduces to

$$W_{h,\,\mathrm{FG}}^{n+1} = \mathcal{G}_{\mathrm{pi-src},\,\mathrm{IRS},\,h}^{(q=5),\,\mathrm{FG}}\circ W_h^n \,. \tag{3.89}$$

3.3.3.2 Restriction Operator

For the data transfer from a fine grid to a coarse grid one has to distinguish two cases: The values of the variables are injected from the fine to the coarse grid only for those points which belong to both grids whereas the residuals needed for the forcing function are restricted to the coarse grid by a volume weighted approach.

The variable values on the fine and on the coarse grid are identical for coinciding points

$$W_{2h,\,i,j}^{n} = \widetilde{I}_h^{2h}\circ W_{h,\mathrm{FG};\,2i,2j}^{n+1} = W_{h,\mathrm{FG};\,2i,2j}^{n+1} \tag{3.90}$$

since these values are injected. The same restriction operator is applied to the turbulence source terms and to their derivative matrices.

For the restriction of residuals a volume weighted approach is used [102]. For the implementation in the FLOWer code it has been assumed that the volumes of the cells

surrounding the point $(2i, 2j)$ on the fine grid are of the same size. Then the restriction operator for the residuals reads in one dimension

$$I_h^{2h}\Big|^i \circ R^{n+1}_{h,\,2i,2j} = \frac{1}{2}R^{n+1}_{h,\,2i-1,2j} + R^{n+1}_{h,\,2i,2j} + \frac{1}{2}R^{n+1}_{h,\,2i+1,2j} \tag{3.91}$$

with the residuals at time level $t + \Delta t$ given as

$$R^{n+1} = (Z_c\circ\, + Z_v\circ\, + Z_S\circ\,)\, W^{n+1}_{\mathrm{FG}}\,. \tag{3.92}$$

Operator notation of Eq. (3.91) yields

$$I_h^{2h}\Big|^i \circ R^{n+1}_{h,\,2i,2j} = \left(\frac{1}{2}\mathcal{E}_i^{-1}\circ\, + \mathcal{I}\circ\, + \frac{1}{2}\mathcal{E}_i\circ\,\right) R^{n+1}_{h,\,2i,2j}\,. \tag{3.93}$$

In analogy the restriction operator in j–direction reads

$$I_h^{2h}\Big|^j \circ\, = \frac{1}{2}\mathcal{E}_j^{-1}\circ\, + \mathcal{I}\circ\, + \frac{1}{2}\mathcal{E}_j\circ\,\,. \tag{3.94}$$

The two dimensional full weighting restriction operator $I_h^{2h}\circ\,$ can be written as the operator product of two one dimensional restriction operators [6]:

$$I_h^{2h}\circ\, = I_h^{2h}\Big|^j \circ\; I_h^{2h}\Big|^i \circ\,\,. \tag{3.95}$$

The lower index of the inter grid transfer operators $\widetilde{I}_*^*\circ\,$ and $I_*^*\circ\,$ denotes the source grid while the upper index stands for the destination grid.

3.3.3.3 Coarse Grid Operator

On the coarse grid the residual is modified by means of a so–called forcing function F_{2h} which is built of transferred fine grid residuals and current coarse grid residuals:

$$\widetilde{R}^n_{2h} = R^n_{2h} + F_{2h} = R^n_{2h} + \left(I_h^{2h}\circ R_h^{n+1} - R^n_{2h}\right)\,. \tag{3.96}$$

By the transferred fine grid residuals incorporated in the forcing function no longer

$$W^{n+1} - W^n = \tau Z\circ W^n \to 0 \tag{3.97}$$

is approached on the coarse grid but

$$W^{n+1}_{2h} - W^n_{2h} = \tau_{2h} Z_{2h}\circ W^n_{2h} \to I_h^{2h}\circ R_h^{n+1}\,. \tag{3.98}$$

Thus the coarse grid solution is modified compared to the fine grid solution in order to reach a coarse grid correction equal to the transferred fine grid residual if time tends to infinity. Transferring back such a coarse grid correction to the fine grid speeds up convergence to steady state on the fine grid without loss of accuracy.

As the forcing function does not depend on the coarse grid time stepping scheme this results in

$$\widetilde{R}^{(q-1)}_{2h} = R^{(q-1)}_{2h} + F_{2h} = R^{(q-1)}_{2h} + \left(I_h^{2h}\circ R_h^{n+1} - R^{(0)}_{2h}\right)\,. \tag{3.99}$$

for each stage residual. Inserting $\widetilde{R}_{2h}^{(q-1)}$ to Eq. (3.55) being applied on a coarse grid yields

$$\begin{aligned} W_{2h}^{(q)} &= W_{2h}^{(0)} + \alpha_q \tau_{2h} \widetilde{R}_{2h}^{(q-1)} \\ &= W_{2h}^{(0)} + \alpha_q \tau_{2h} R_{2h}^{(q-1)} + \alpha_q \tau_{2h} F_{2h} \end{aligned} \tag{3.100}$$

as the qth Runge–Kutta stage on the coarse grid. With point implicit source term treatment (Sect. 3.2.2.2) and implicit residual smoothing according to Sect. 3.3.2

$$\begin{aligned} f_{\mathrm{pi-src,\,IRS},\,2h}^{(q)} \circ &= B_{2h}^{-1} \circ \Bigg[\alpha_{q,\,2h}^{\mathrm{pi-src}} + \alpha_{q,\,2h}^{\mathrm{pi-src}} \alpha_{q-1,\,2h}^{\mathrm{pi-src}} \tau_{2h} Z_{c,\,2h} \circ B_{2h}^{-1} \circ \\ &\quad + \ldots + \prod_{r=1}^{q} \alpha_{r,\,2h}^{\mathrm{pi-src}} \left(\tau_{2h} Z_{c,\,2h} \circ B_{2h}^{-1} \circ \right)^{q-1} \Bigg] \end{aligned} \tag{3.101}$$

with

$$\alpha_{q,\,2h}^{\mathrm{pi-src}} = \frac{\alpha_q}{1 - \alpha_q \tau_{2h} \dfrac{\partial \left(Z_{S,\,2h} \circ W^{(0)} \right)}{\partial W}} \tag{3.102}$$

is gained as the characterizing operator polynomial of the qth stage of the Runge–Kutta scheme on the coarse grid by using $\widetilde{R}^{(q)}$ instead of $R^{(q)}$. The differences in the derivation are briefly presented in Sect. A.3.3. This characterizing polynomial equals Eq. (3.83) with all operators and τ defined on the coarse grid, i. e. the time stepping scheme is independent of the forcing function.

Using Eq. (3.101) and Eq. (3.96) the value of the variable on the coarse grid is calculated from the fine grid values due to the forcing function as

$$\begin{aligned} W_{2h}^{n+1} &= W_{2h}^{n} + f_{\mathrm{pi-src,\,IRS},\,2h}^{(q=5)} \circ \tau_{2h} \widetilde{R}_{2h}^{n} \\ &= W_{2h}^{n} + f_{\mathrm{pi-src,\,IRS},\,2h}^{(q=5)} \circ \tau_{2h} \left(Z_{2h} \circ \widetilde{I}_{h}^{2h} \circ + F_{2h} \circ \right) W_{h,\,FG}^{n+1} \\ &= \left(\widetilde{I}_{h}^{2h} \circ + f_{\mathrm{pi-src,\,IRS},\,2h}^{(q=5)} \circ \tau_{2h} \left[Z_{2h} \circ \widetilde{I}_{h}^{2h} \circ + F_{2h} \circ \right] \right) W_{h,\,FG}^{n+1} . \end{aligned} \tag{3.103}$$

Here the forcing function F_{2h} has been developed to a corresponding operator

$$F_{2h} \circ W_{h,\,FG}^{n+1} = \left(I_{h}^{2h} \circ Z_{h} \circ - Z_{2h} \circ \widetilde{I}_{h}^{2h} \circ \right) W_{h,\,FG}^{n+1} . \tag{3.104}$$

The last equality in Eq. (3.103) is related to the definition of the restriction operator for the fine grid variables, Eq. (3.90), in the previous section. Thus the operator of the Runge–Kutta time stepping scheme on the coarse grid is derived as

$$\mathcal{G}_{\mathrm{pi-src,\,IRS},\,2h}^{(q=5),\,\mathrm{CG}} \circ = \widetilde{I}_{h}^{2h} \circ + f_{\mathrm{pi-src,\,IRS},\,2h}^{(q=5)} \circ \tau_{2h} \left[Z_{2h} \circ \widetilde{I}_{h}^{2h} \circ + F_{2h} \circ \right] \tag{3.105}$$

which has to be applied to fine grid data. The superscript ^{CG} denotes that this operator belongs to a coarse grid within a multigrid cycle.

3.3.3.4 Prolongation Operator

The data to be transferred from the coarse to the fine grid is the so–called coarse grid correction. This is the difference between the starting value on the coarse grid and the value after one (or more) time step(s):

$$\begin{aligned} \Delta W_{2h} &= W_{2h}^{n+1} - W_{2h}^{n} \\ &= \left(\mathcal{G}_{\mathrm{pi-src,\,IRS,\,2h}}^{(q=5),\,\mathrm{CG}} \circ \; - \widetilde{I}_{h}^{2h} \circ \right) W_{h,FG}^{n+1} \,. \end{aligned} \tag{3.106}$$

Here the identity of coarse grid variable values with fine grid variable values for coinciding points, Eq. (3.90), is used together with Eq. (3.105) for the operator of the coarse grid time stepping scheme.

Two multigrid levels of a two dimensional grid are sketched in Fig. 3.5. The thick lines are coarse (and fine) grid lines while the dashed lines indicate the fine grid. The indices i and j belong to the coarse grid. The transfer of the coarse grid corrections to the fine grid has to distinguish three cases for a two dimensional grid:

1. The coarse grid point coincides with a fine grid point: $(i, j) \mapsto (2i, 2j)$ (bullets in Fig. 3.5)
2. The fine grid point is located on the center of a coarse grid cell face:
   (a) The fine grid point is located on the center of a coarse grid cell face between the coarse grid indices (i, j) and $(i+1, j)$: $(2i+1, 2j)$ (open circles in Fig. 3.5)
   (b) The fine grid point is located on the center of a coarse grid cell face between the coarse grid indices (i, j) and $(i, j+1)$: $(2i, 2j+1)$ (open circles in Fig. 3.5)
3. The fine grid point resides in the center of a coarse grid cell: $(2i+1, 2j+1)$ (thick open circles in Fig. 3.5)

A generic prolongation of coarse grid values to fine grid points consists of two combined steps: At first the values at coarse grid points are transferred to the coinciding fine grid points (list item no. 1). Secondly the values at fine grid points non–coincident with coarse grid points have to be calculated (list item no. 2 and no. 3).
The first step is performed by means of an injection of the coarse grid correction:

$$\Delta W_{h,\,2i,2j} = \widetilde{I}_{2h}^{h} \circ \Delta W_{2h,\,i,j} = c_{i,j} \Delta W_{2h,\,i,j} \,. \tag{3.107}$$

In Fig. 3.6 the letter "c" located close to the bullets of the coarse grid points denote the relaxation factor used for the injection of coarse grid values.

During the second step the intermediate fine grid points are calculated from surrounding coarse grid values which have been injected to the fine grid before. For an in–between value in i–direction (cf. list item no. 2a) this results in

$$\begin{aligned} \Delta W_{h,\,2i+1,2j} &= I_{2h}^{h}\Big|^{i} \circ \Delta W_{2h,\,i+\frac{1}{2},j} \\ &= I_{[2]h}^{h}\Big|^{i} \circ \left(\widetilde{I}_{2h}^{h} \circ \Delta W_{2h,\,i+\frac{1}{2},j} \right) \,. \end{aligned} \tag{3.108}$$

$(2h,\ i+1/2, j)$ is just a virtual point on the coarse grid necessary for the notation of the prolongation operator. The operator $I^h_{[2]h}\Big|^i \circ$ is a (linear) interpolation operator acting on fine grid values only. Its subscript contains "[2]" to denote that it is always part of a prolongation operator. It is defined as

$$I^h_{[2]h}\Big|^i \circ \Delta W_{h,\,2i+1,2j} = a^- \Delta W_{h,\,2i,2j} + a^+ \Delta W_{h,\,2i+2,2j} \tag{3.109}$$

if only the closest neighbors are considered.
For the computation of the in–between value Eq. (3.109) is used, i. e. in the FLOWer code

$$\Delta W_{h,\,2i+1,2j} = a^- \Delta W_{h,\,2i,2j} + a^+ \Delta W_{h,\,2i+2,2j} \tag{3.110}$$

is implemented based on coarse grid corrections already injected to the fine grid according to Eq. (3.107).

Another sweep over the j–direction is used to calculate values at intermediate grid points in j–direction (cf. list items no. 2b and no. 3).
The arrows in Fig. 3.6 show which coarse grid values — after being injected to fine grid points — and which coefficients are used for calculating fine grid values.

Using the spatial index shift operator $\mathcal{E}\circ$ the prolongation operator for the i–direction — including the injection of coarse grid values — reads

$$I^h_{2h}\Big|^i \circ \;=\; I^h_{[2]h}\Big|^i \circ \widetilde{I}^h_{2h} \circ \;=\; \left(a^- \mathcal{E}_i^{-1} \circ \;+ a^+ \mathcal{E}_i \circ\right) \widetilde{I}^h_{2h} \circ \;. \tag{3.111}$$

The analogous operator for the j–direction is given as

$$I^h_{2h}\Big|^j \circ \;=\; I^h_{[2]h}\Big|^j \circ \widetilde{I}^h_{2h} \circ \;=\; \left(b^- \mathcal{E}_j^{-1} \circ \;+ b^+ \mathcal{E}_j \circ\right) \widetilde{I}^h_{2h} \circ \;. \tag{3.112}$$

For the fine grid points which are defined by list item no. 3 the prolongation operator equals the product of the two one–dimensional prolongation operators and the injection operator:

$$I^h_{2h} \circ \;=\; I^h_{[2]h}\Big|^j \circ \;\; I^h_{[2]h}\Big|^i \circ \widetilde{I}^h_{2h} \circ \;. \tag{3.113}$$

In the following the prolongation operator is always denoted as $I^h_{2h}\circ$ independently of the destination fine grid point.

The default prolongation operator of the FLOWer code is an unrelaxed bilinear interpolation operator. The operator is defined by choosing

$$a^- = a^+ = b^- = b^+ = \frac{1}{2} \qquad \text{and} \qquad c = 1.0 \tag{3.114}$$

in Eq. (3.113). It is visualized in Fig. 3.7.

3.3.3.5 Operator of the Two Grid Method

The coarse grid corrections ΔW_{2h} are subject to an implicit smoothing ($B_{\mathrm{MG}}\circ$) after being prolongated to the fine grid as noted above. Thus the value of the variable on the fine grid after a two grid cycle reads

$$W_h^{t+\Delta t,\,\mathrm{MG}} = W_{h,\,\mathrm{FG}}^{n+1} + B_{\mathrm{MG}}^{-1} \circ I^h_{2h} \circ \Delta W_{2h}\,. \tag{3.115}$$

Inserting Eq. (3.106) for ΔW_{2h} results in

$$W_h^{t+\Delta t,\,\mathrm{MG}} = \left(\mathcal{I}\circ + B_{\mathrm{MG}}^{-1}\circ I_{2h}^{h}\circ \left[\mathcal{G}_{\mathrm{pi-src,\,IRS},\,2h}^{(q=5),\,\mathrm{CG}}\circ - \widetilde{I}_{h}^{2h}\circ\right]\right) W_{h,\,\mathrm{FG}}^{n+1}\,. \tag{3.116}$$

Using Eq. (3.89) for $W_{h,\,\mathrm{FG}}^{n+1}$ yields

$$W_h^{t+\Delta t,\,\mathrm{MG}} = \mathcal{G}_{\mathrm{pi-src,\,IRS}}^{(q=5),\,2\mathrm{G}}\circ W_h^n \tag{3.117}$$

with

$$\mathcal{G}_{\mathrm{pi-src,\,IRS}}^{(q=5),\,2\mathrm{G}}\circ = \left(\mathcal{I}\circ + B_{\mathrm{MG}}^{-1}\circ I_{2h}^{h}\circ \left[\mathcal{G}_{\mathrm{pi-src,\,IRS},\,2h}^{(q=5),\,\mathrm{CG}}\circ - \widetilde{I}_{h}^{2h}\circ\right]\right)\mathcal{G}_{\mathrm{pi-src,\,IRS},\,h}^{(q=5),\,\mathrm{FG}}\circ \tag{3.118}$$

for the operator of a two level multigrid scheme.

3.3.3.6 Operator of the Multi Grid Method

The following derivation has already been given by Jameson in [51] but in a less general context.
For a multigrid scheme with an arbitrary number of grid levels the coarse grid corrections $\Delta W_{2^{\mathtt{level}-1}h}$ of each level `level` have to be calculated and prolongated to the finest grid where they are subject to an implicit smoothing operation ($B_{\mathrm{MG}}^{-1}\circ$) as in the two grid cycle. Thus the value of the variable on the finest grid after a multigrid cycle reads

$$W_h^{t+\Delta t,\,\mathrm{MG}} = W_{h,\,\mathrm{FG}}^{n+1} + B_{\mathrm{MG}}^{-1}\circ I_{2h}^{h}\circ \left(\Delta W_{2h} + I_{4h}^{2h}\circ\left(\Delta W_{4h} + \ldots\right)\right)\,. \tag{3.119}$$

In order to calculate ΔW_{4h} a corresponding coarse grid operator is needed. Because of the forcing function the residual on the third level reads

$$\begin{aligned}
\widetilde{R}_{4h}^{n} &= R_{4h}^{n} + F_{4h} \\
&= R_{4h}^{n} + \left(I_{2h}^{4h}\circ R_{2h}^{n+1} - R_{4h}^{n} + I_{2h}^{4h}\circ F_{2h}\right) \\
&= R_{4h}^{n} + \left(I_{2h}^{4h}\circ R_{2h}^{n+1} - R_{4h}^{n} + I_{2h}^{4h}\circ\left[I_{h}^{2h}\circ R_{h}^{n+1} - R_{2h}^{n}\right]\right)\,.
\end{aligned} \tag{3.120}$$

From this a recursive form of Eq. (3.104), the definition of the operator corresponding to the forcing function, can be derived as

$$F_{2H}\circ = \left[I_{H}^{2H}\circ Z_{H}\circ - Z_{2H}\circ\widetilde{I}_{H}^{2H}\circ\right]\mathcal{G}_{\mathrm{pi-src,\,IRS},\,H}^{(q=5),\,\mathrm{CG}}\circ + I_{H}^{2H}\circ F_{H}\circ \tag{3.121}$$

using $H \stackrel{\mathrm{def}}{=} 2^{\mathtt{level}-1}h$ for any multigrid level `level`. This equation shows why the forcing function introduces the fine grid influence to any coarse grid level.
From Eq. (3.120) some transformations analogous to those in Eq. (3.103) yield the coarse grid operator for the third level

$$\begin{aligned}
\mathcal{G}_{\mathrm{pi-src,\,IRS},\,4h}^{(q=5),\,\mathrm{CG}}\circ &= \widetilde{I}_{2h}^{4h}\circ\mathcal{G}_{\mathrm{pi-src,\,IRS},\,2h}^{(q=5),\,\mathrm{CG}}\circ \\
&\quad + f_{\mathrm{pi-src,\,IRS},\,4h}^{(q)}\circ\tau_{4h}\left(Z_{4h}\circ\widetilde{I}_{2h}^{4h}\circ\mathcal{G}_{\mathrm{pi-src,\,IRS},\,2h}^{(q=5),\,\mathrm{CG}}\circ + F_{4h}\circ\right)
\end{aligned} \tag{3.122}$$

which has to be applied to fine grid values like the coarse grid operator of the second grid level as noted in Sect. 3.3.3.3. Eq. (3.122) shows that the time stepping operators on successively coarser grids can be defined recursively.

From Eq. (3.122) it can be concluded easily that ΔW_{4h} is given by

$$\begin{aligned} \Delta W_{4h} &= W_{4h}^{n+1} - W_{4h}^{n} \\ &= \mathcal{G}^{(q=5),\,\mathrm{CG}}_{\mathrm{pi-src,\,IRS},\,4h} \circ W_{h,\,FG}^{n+1} - \widetilde{I}^{4h}_{2h} \circ W_{2h}^{n+1} \\ &= \left(\mathcal{G}^{(q=5),\,\mathrm{CG}}_{\mathrm{pi-src,\,IRS},\,4h} \circ \; - \widetilde{I}^{4h}_{2h} \circ \mathcal{G}^{(q=5),\,\mathrm{CG}}_{\mathrm{pi-src,\,IRS},\,2h} \circ \right) W_{h,\,FG}^{n+1} \,. \end{aligned} \tag{3.123}$$

Inserting Eq. (3.106) for ΔW_{2h}, Eq. (3.123) for ΔW_{4h} and corresponding formulae for other coarse grid corrections in Eq. (3.119) as well as using Eq. (3.88) for operator notation of $W_{h,\,\mathrm{FG}}^{n+1}$ yields

$$\begin{aligned} \mathcal{G}^{(q=5),\,\mathrm{MG}}_{\mathrm{pi-src,\,IRS}} \circ \;\; &= \Big(\mathcal{I} \circ \; + B_{\mathrm{MG}}^{-1} \circ I_{2h}^{h} \circ \left[\mathcal{G}^{(q=5),\,\mathrm{CG}}_{\mathrm{pi-src,\,IRS},\,2h} \circ \; - \widetilde{I}^{2h}_{h} \circ \right] \\ &\quad + B_{\mathrm{MG}}^{-1} \circ I_{2h}^{h} \circ I_{4h}^{2h} \circ \left[\mathcal{G}^{(q=5),\,\mathrm{CG}}_{\mathrm{pi-src,\,IRS},\,4h} \circ \; - \widetilde{I}^{4h}_{2h} \circ \mathcal{G}^{(q=5),\,\mathrm{CG}}_{\mathrm{pi-src,\,IRS},\,2h} \circ \right] \\ &\quad + \ldots \Big) \mathcal{G}^{(q=5),\,\mathrm{FG}}_{\mathrm{pi-src,\,IRS},\,h} \circ \end{aligned} \tag{3.124}$$

for the operator of a multi level multigrid scheme. This operator denotes a sawtooth V–cycle with one Runge–Kutta cycle on each level while sweeping from the finest to the coarsest grid level and no Runge–Kutta step while sweeping back.

Different kinds of cycles or additional Runge–Kutta steps while sweeping back to the finest grid would change this operator significantly as the coarse grid corrections need to be redefined in such a case. Because of the complexity of these modifications this part of the work is restricted to a sawtooth V–cycle.

3.3.3.7 Limiting of Turbulence Variables and Related Values in Multigrid Method

As noted in Sect. 3.2.3, whenever the turbulence quantities are updated, one has to ensure that those variables are positive. This procedure is performed on each Runge–Kutta stage on all multigrid levels in the same way. However, two more situations within a multigrid cycle require a lower bound of the turbulence variables: At first the coarse grid corrections need to be bounded after prolongation to the next finer mesh:

$$\Delta_H W \;=\; I_{2H}^{H} \circ \Delta W_{2H} \,. \tag{3.125a}$$

This is done by calculating pseudo updated turbulence variables

$$W_{H;\,\mathrm{pseudo}} \;=\; W_H + \Delta_H W \,, \tag{3.125b}$$

bounding these

$$W_{H;\,\mathrm{pseudo;\,lim}} \;=\; \max\left(W_{H;\,\mathrm{pseudo}}, C_{W;\mathrm{lim}} \cdot W_\infty \right) \tag{3.125c}$$

and recalculating bounded coarse grid corrections as

$$\Delta_{H;\,\mathrm{lim}} W \;=\; W_{H;\,\mathrm{pseudo;\,lim}} - W_H \,. \tag{3.125d}$$

Secondly the question has to be answered how the positivity of the turbulence quantities is to be assured at the end of a multigrid cycle. Although the bounding of coarse grid corrections can be performed on the finest grid as well here the bounding is done in the same way as on a Runge–Kutta stage: The turbulence variables on the finest grid are updated by adding the interpolated coarse grid corrections. Afterwards the Eqs. (3.71) and (3.72) are applied to these multigrid–updated turbulence quantities.

4 Fourier Analysis of the Numerical Treatment of Turbulence Equations

In order to get prepared for a deeper insight into the problems and phenomena to be described in Chap. 5, a stability analysis according to von Neumann is introduced. Von Neumann analysis is a standard tool to investigate the (damping) properties of a solution algorithm for partial differential equations [128, 114], and is also used extensively for the flow equations [46, 54, 6].

The von Neumann analysis is based on a Fourier analysis in space. Assuming periodic boundary conditions, the residual of a partial differential equation — which is restricted to be linear — is developed in a finite series of Fourier modes. The influence of a numerical method on this Fourier series is investigated. On the one hand this yields information about the stability of the solution algorithm. On the other hand a detailed description of the behaviour of all Fourier modes of the computational error is gained. The behaviour of the Fourier modes is important for estimating the convergence quality of a solution method as the amplitude of each error mode is to be reduced (damped) as fast as possible.
At least this "local mode analysis" answers the question how any mode in an arbitrary but fixed grid point changes over one iteration cycle of the numerical method applied. It is described in a more theoretical way in [57, 114, 10].
Within this work coupling of modes as introduced inherently by multigrid methods is not taken into account in accordance with the works of Jameson [51], Radespiel et al. [98] and Blazek [6]. Other authors [48, 69, 11, 134] put emphasis on this coupling especially in the context of investigating systems of equations.

As the von Neumann analysis is only applicable to linear problems the partial differential equations under investigation need to be approximated by a suitable linear model. While convection equations, diffusion equations or combination of both are well known model problems for the von Neumann analysis [129, 63, 134], in this work the linear model equation is to approximate a turbulence model equation, i. e. a convection–diffusion equation with a source term, in two dimensions. Such a model is introduced in Sect. 4.1.
In this context Hirsch [46] points out that "linear stability is a necessary condition for non–linear problems but it is certainly not sufficient". As an example aliasing of error modes caused by the nonlinearity of an equation cannot be analyzed within a linear model problem.

One word on the nomenclature: The term "Fourier symbol" is frequently used [63, 128, 48, 51, 69, 94, 134] for a Fourier transformed discretization operator and it will be used here as well. A more rigorous term would be "spectral function" as used in [128].

In Sect. 4.2 all Fourier symbols needed are derived from the semi–discretizations given in Sect. 3.1 using a consistent operator notation. This operator notation is unlike the Fourier analyses performed by other authors [6, 7, 46, 63, 94, 129, 134]. It shows the close relation between the operators of the numerical method and of the Fourier analysis. Above all this operator notation allows for a consistent formulation ranging from derivation to analysis of a numerical method.
Subsequently the spatial Fourier symbols are embedded in the Fourier analysis of the time stepping scheme (Sect. 4.3). In Sect. 4.4 the achieved Fourier symbol is extended by using implicit residual smoothing and multigrid. Usually two grid levels are used for multigrid analysis only [6, 48, 98, 100, 129, 134]. In order to be able to fully capture the effects of source terms within a multigrid method here a Fourier symbol for an arbitrary number of multigrid levels is derived.
Some basic results of this Fourier analysis are presented in Sect. 4.5. Special emphasis is laid on the derivation of realistic settings for the Fourier analysis. Sect. 4.6 gives the conclusions from this chapter.

4.1 Description of Model Problem

The linear model problem used for the von Neumann analysis of a turbulence transport equation is introduced and discussed in this section. Moreover the necessary assumptions are given.

As a basis of the Fourier analysis it is assumed that the grid on which the model problem is to be discretized is finite and has rectangular cells with equidistant point distributions in each coordinate direction. In the model problem periodic boundary conditions are presupposed. Because of this the same discretization stencil can be used in every point of the grid. In order to apply the principle of superposition to the Fourier modes, linearity of the model problem is necessary.

One major question has to be answered before searching for a model problem approximating the turbulence equations: Periodic boundary conditions are normally applied in order to remove the influence of the boundary conditions on the equations under investigation. Is the influence of the boundary conditions likewise negligible, or removable for turbulence equations?
The answer is ”Yes”. This can be traced to the character of the turbulence equations which are transport equations with source terms: Combinations of convection and diffusion terms are well known and frequently used in model problems for the Fourier analysis of solution algorithms for the Euler or Navier–Stokes equations [6, 97, 129]. As the source term depends on velocity gradients and on local values of the turbulence variables it does not depend on boundary conditions for the turbulence quantities more than the convective or diffusive terms do. Therefore the assumption of negligible boundary influence is tenable also for an approximation to turbulence equations as it is for an approximation to the RANS equations.
One should keep in mind that this neglect is not valid globally neither for convection and diffusion terms nor for source terms.

The model problem is restricted to a scalar equation instead of a system of equations since the numerical treatment of the turbulence equations as introduced in Chap. 3

does not account for interdependence either. Work related to mutual dependencies has been done e. g. by Ibraheem et al. [48] or Kunz et al. [69]. While Ibraheem and Demuren do not compare their results for systems of equations with corresponding results for scalar equations, Kunz and Lakshminarayana state that they found no differences in the stability results obtained for a coupled system of RANS and turbulence equations and for an uncoupled system of these equations.
For the Fourier analysis a linear model problem is needed. The scalar equation used for the introduction of the numerical method,

$$\iint_V \frac{\partial}{\partial t} W \,\mathrm{d}V = -\int_{\partial V} \bar{\bar{F}} \bullet \vec{n}\,\mathrm{d}(\partial V) + \iint_V S\,(W)\;\mathrm{d}V\,, \tag{2.53}$$

is rewritten as a linear equation in differential formulation as follows:

$$\frac{\partial W}{\partial t} = -u\frac{\partial W}{\partial x} - v\frac{\partial W}{\partial y} + \mu\frac{\partial^2 W}{\partial y^2} + S \cdot W\,. \tag{4.1}$$

Constant, positive velocities u and v are assumed for the Fourier analysis in order to simplify the convective terms. The viscous terms are reduced by the same assumption on the viscosity coefficient μ. Additionally, a thin shear layer approximation of the viscous terms is taken into account as this is the standard approach in FLOWer. The source term is approximated in Eq. (4.1) by a linear function. Thus S denotes a constant source term coefficient.

4.2 Fourier Symbols of Spatial Discretization Operators

In this section the Fourier symbols of the spatial discretization of the model problem are derived. These Fourier symbols (also called *spectral functions*) represent the influence of the spatial discretization on the time development of the amplitudes of the Fourier modes. Thus Eq. (4.1) is rewritten as

$$\frac{\partial \widehat{W}}{\partial t} = \mathcal{Z} \cdot \widehat{W} \tag{4.2}$$

by means of Fourier transformation. $\widehat{W}$ denotes the amplitude of the mode being investigated and $\mathcal{Z}$ stands for the Fourier symbols of the spatial discretization operators. The operator notation according to Vichnevetsky and Bowles [128] as introduced in Sect. 3.1 and subsequent sections of Chap. 3 is used here again. This operator notation is well adapted for the calculation of the amplification factor of the full discretization and its extension to implicit residual smoothing and multigrid, as done in Sect. 4.3 and Sect. 4.4.

The Fourier symbol for the semi–discretization of the model problem equation, Eq. (4.1), will be derived term by term for reasons of readability: First a 2D convection equation is used (Sect. 4.2.1). In the same section the applicability of the operators derived in Chap. 3 to the model problem, Eq. (4.1), is substantiated. Afterwards its Fourier symbol is extended by including the viscous term (Sect. 4.2.2). In a last step the source term is included in the Fourier symbol (Sect. 4.2.3).

Depending on the discretization analogous work has been done e. g. by Kroll and Jain [63], Blazek [6, 7] or Radespiel and Swanson [98] but much more briefly. Here a formulation of the Fourier symbols is presented which is consistent with the notation of the discretization operators themselves.

4.2.1 Fourier Symbol of Inviscid Terms

From the assumptions of rectangular cells and equidistantly distributed points one gets

$$\mathcal{C}_{i+\frac{1}{2},j} = \mathcal{C}_{i-\frac{1}{2},j} = \Delta y \quad \text{and} \quad \mathcal{C}_{i,j+\frac{1}{2}} = \mathcal{C}_{i,j-\frac{1}{2}} = \Delta x \tag{4.3}$$

for the cell face areas. The cell face normal unit vectors — compare also Fig. 3.4 resp. Fig. 3.3 — are

$$\vec{n}_{i+\frac{1}{2},j} = \vec{n}_{i-\frac{1}{2},j} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad \vec{n}_{i,j+\frac{1}{2}} = \vec{n}_{i,j-\frac{1}{2}} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} . \tag{4.4}$$

From Eq. (4.3) the volume of the control cell $V_{i,j}$ is derived as

$$\mathcal{V}_{i,j} = \Delta x \Delta y \, . \tag{4.5}$$

By assuming constant positive velocities, $\vec{q} = (u, v)^{\mathsf{T}}$, in the following the velocities are given by

$$q_{\vec{n};\, i+\frac{1}{2},j} = q_{\vec{n};\, i-\frac{1}{2},j} = u \quad \text{and} \quad q_{\vec{n};\, i,j+\frac{1}{2}} = q_{\vec{n};\, i,j-\frac{1}{2}} = v \tag{4.6}$$

From Eq. (3.23) it follows in addition that

$$|q|_{\vec{n};\, i+\frac{1}{2},j} = |q|_{\vec{n};\, i-\frac{1}{2},j} = u \quad \text{and} \quad |q|_{\vec{n};\, i,j+\frac{1}{2}} = |q|_{\vec{n};\, i,j-\frac{1}{2}} = v \, . \tag{4.7}$$

Applying these simplifying assumptions to the operator $Z_c|^i \circ$, Eq. (3.26) reads

$$Z_c|^i \circ \; = -u \Delta y \left(\mathcal{I} \circ \; - \mathcal{E}_i^{-1} \circ \right) \tag{4.8}$$

Using this operator together with its analogue for the convection in j–direction — Eq. (3.27) —,

$$Z_c|^j \circ \; = -v \Delta x \left(\mathcal{I} \circ \; - \mathcal{E}_j^{-1} \circ \right) , \tag{4.9}$$

one arrives at

$$\begin{aligned}
-u \left.\frac{\partial W}{\partial x}\right|_{i,j} - v \left.\frac{\partial W}{\partial y}\right|_{i,j} &\approx -\frac{u}{\Delta x} \left(W_{i,j} - W_{i-1,j} \right) - \frac{v}{\Delta y} \left(W_{i,j} - W_{i,j-1} \right) && \text{(4.10a)} \\
&= \frac{1}{\Delta x \Delta y} \left[-u \Delta y \left(W_{i,j} - W_{i-1,j} \right) - v \Delta x \left(W_{i,j} - W_{i,j-1} \right) \right] && \text{(4.10b)} \\
&= \frac{1}{\mathcal{V}_{i,j}} \left[-u \Delta y \left(\mathcal{I} \circ \; - \mathcal{E}_i^{-1} \circ \right) - v \Delta x \left(\mathcal{I} \circ \; - \mathcal{E}_j^{-1} \circ \right) \right] W_{i,j} && \text{(4.10c)} \\
&= \frac{1}{\mathcal{V}_{i,j}} \left[Z_c|^i \circ \; + \; Z_c|^j \circ \right] W_{i,j} && \text{(4.10d)}
\end{aligned}$$

as a discrete approximation to the convection of the model problem Eq. (4.1) in grid point (i, j). The right hand side of Eq. (4.10a) denotes a first order upwind discretization of the convective terms as given e. g. in [1, 46]. The factor $1/\mathcal{V}_{i,j}$ in Eq. (4.10d) is

already known from Eq. (3.1) and will be used again (cmp. Sect. 3.2.1) to define the time step per control volume for the time stepping scheme in Sect. 4.3.

Assuming periodic boundary conditions the discrete and finite Fourier representation of $W_{i,j}$ is given by

$$W_{i,j} = \sum_{l_x,l_y=-N}^{N} \widehat{W}_{l_x,l_y} e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y} \ . \tag{4.11}$$

Here $\widehat{W}_{l_x,l_y}$ denotes the amplitude of the (l_x, l_y)th Fourier mode. $\mathbf{I}$ is the imaginary unit whereas $i, j \in \{0, 1, \ldots, N\}$ are the grid point indices. Thus N is the number of points taken to resolve the computational space of the model problem in x– as well as in y–direction. Φ_x and Φ_y stand for the phase angles and are defined as

$$\Phi_x = \frac{l_x \pi}{N} \quad \text{and} \qquad \Phi_y = \frac{l_y \pi}{N} \, . \tag{4.12}$$

As the linearity of the model equation allows superposition of modes the investigation of a single but general mode leads to results valid for all modes. In this analysis $W_{i,j}$ is therefore replaced by

$$W_{i,j} \rightarrow \widehat{W} e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y} \ . \tag{4.13}$$

Inserting this replacement into the right hand side of Eq. (4.10d) yields

$$Z_c \circ e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y} \widehat{W} \tag{4.14}$$

which is a Fourier representation of the spatially discretized convection terms. According to Vichnevetsky and Bowles [128] the Fourier symbol $\mathcal{Z}(\Phi_x, \Phi_y)$ of a semi–discretization is defined as the ratio between the Fourier representations of the spatial discretizations of the right and left hand side. As the spatial discretization operator of the left hand side of Eq. (4.1), i. e. the time derivative, equals the identity one gets

$$\mathcal{Z}_c(\Phi_x, \Phi_y) = \frac{Z_c \circ e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y}}{\mathcal{I} \circ e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y}} \tag{4.15}$$

for the Fourier symbol of the convective part of Eq. (4.1). Inserting the formula for $Z_c \circ$ derived above into Eq. (4.15) leads to

$$\begin{aligned} \mathcal{Z}_c(\Phi_x, \Phi_y) &= -u\Delta y \frac{e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y} - e^{\mathbf{I}(i-1)\Phi_x + \mathbf{I}j\Phi_y}}{e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y}} \\ &\quad - v\Delta x \frac{e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y} - e^{\mathbf{I}i\Phi_x + \mathbf{I}(j-1)\Phi_y}}{e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y}} \\ &= -u\Delta y \left[1 - e^{-\mathbf{I}\Phi_x}\right] - v\Delta x \left[1 - e^{-\mathbf{I}\Phi_y}\right] \, . \end{aligned} \tag{4.16}$$

Rewriting this result by means of trigonometric functions gives

$$\mathcal{Z}_c(\Phi_x, \Phi_y) = -u\Delta y \left[1 - \cos(\Phi_x) + \mathbf{I}\sin(\Phi_x)\right] - v\Delta x \left[1 - \cos(\Phi_y) + \mathbf{I}\sin(\Phi_y)\right] \tag{4.17}$$

for the Fourier symbol of the spatial discretization operator of the convection terms of the model equation Eq. (4.1).

4.2.2 Fourier Symbol of Viscous Terms

As a second step the viscous term of the model problem equation, Eq. (4.1), is taken into account: At first those terms of the discretization operator of the viscous term, Eq. (3.41), are discussed which are related to geometrical quantities i. e.

$$(\mathcal{C}\vec{n})^*_{i,j+\frac{1}{2}} \bullet \vec{n}_{i,j+\frac{1}{2}}\, \mathcal{C}_{i,j+\frac{1}{2}} \qquad \text{resp.} \qquad (\mathcal{C}\vec{n})^*_{i,j-\frac{1}{2}} \bullet \vec{n}_{i,j-\frac{1}{2}}\, \mathcal{C}_{i,j-\frac{1}{2}}\,. \tag{4.18}$$

$\mathcal{C}_{i,j\pm\frac{1}{2}}$ and $\vec{n}_{i,j\pm\frac{1}{2}}$ are given by Eqs. (4.3) and (4.4). Still the definition of the normal unit vectors $\vec{n}_{I,J\pm\frac{1}{2}}$ and of the cell face areas $\mathcal{C}_{I,J\pm\frac{1}{2}}$ in the framework of this Fourier analysis is needed for the calculation of $(\mathcal{C}\vec{n})^*_{i,j\pm\frac{1}{2}}$, Eq. (3.36). Due to the assumptions of rectangular cells and equidistant point distributions made for the Fourier analysis the equalities

$$\vec{n}_{I,J\pm\frac{1}{2}} = \vec{n}_{i,j\pm\frac{1}{2}} \tag{4.19}$$

and

$$\mathcal{C}_{I,J\pm\frac{1}{2}} = \mathcal{C}_{i,j\pm\frac{1}{2}} \tag{4.20}$$

hold. Analogously the volume of the auxiliary cell $\mathcal{V}_{i,j+\frac{1}{2}}$ equals the volume of one grid cell i. e.

$$\mathcal{V}_{i,j+\frac{1}{2}} = \Delta x \Delta y\,. \tag{4.21}$$

Applying these considerations to Eq. (4.18) yields

$$(\mathcal{C}\vec{n})^*_{i,j\pm\frac{1}{2}} \bullet \vec{n}_{i,j\pm\frac{1}{2}}\, \mathcal{C}_{i,j\pm\frac{1}{2}} = \frac{\Delta x}{\Delta x \Delta y} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \bullet \begin{pmatrix} 0 \\ 1 \end{pmatrix} \Delta x = \frac{\Delta x}{\Delta y}\,. \tag{4.22}$$

In addition to this a constant viscosity coefficient μ is presupposed. One arrives easily at the discretization operator of the viscous term, Eq. (3.41),

$$Z_v \circ \; = \mu \frac{\Delta x}{\Delta y} \left(\mathcal{E}_j \circ \; - 2\mathcal{I} \circ \; + \mathcal{E}_j^{-1} \circ \right)\,. \tag{4.23}$$

The factor $1/\mathcal{V}_{i,j}$ — which is as before used for the definition of the time step per control volume — needs to be multiplied by $Z_v \circ$ in order to gain a correct approximation of the viscous term in the differential formulation:

$$\mu \frac{\partial^2 W}{\partial y^2} \approx \frac{1}{\mathcal{V}_{i,j}} Z_v \circ W_{i,j}\,. \tag{4.24}$$

Inserting $Z_v \circ$ into Eq. (4.15) instead of $Z_c \circ$ for the calculation of the corresponding Fourier symbol results in

$$\begin{aligned} \mathcal{Z}_v(\Phi_x, \Phi_y) &= \mu \frac{\Delta x}{\Delta y} \frac{e^{\mathbf{I}i\Phi_x + \mathbf{I}(j+1)\Phi_y} - 2e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y} + e^{\mathbf{I}i\Phi_x + \mathbf{I}(j-1)\Phi_y}}{e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y}} \\ &= \mu \frac{\Delta x}{\Delta y} \left(e^{\mathbf{I}\Phi_y} - 2 + e^{-\mathbf{I}\Phi_y} \right)\,. \end{aligned} \tag{4.25}$$

Using trigonometric functions one can write the Fourier symbol of the spatial discretization of the viscous term of the model problem as

$$\mathcal{Z}_v(\Phi_x, \Phi_y) = -2\mu \frac{\Delta x}{\Delta y} \left[1 - \cos(\Phi_y) \right]\,. \tag{4.26}$$

The Fourier symbol derived thus far corresponds to a convection diffusion equation (without a source term) reads

$$\begin{aligned}\mathcal{Z}(\Phi_x,\Phi_y) &= \mathcal{Z}_c(\Phi_x,\Phi_y)+\mathcal{Z}_v(\Phi_x,\Phi_y)\\ &= -u\Delta y\left[1-\cos(\Phi_x)+\mathbf{I}\sin(\Phi_x)\right]-v\Delta x\left[1-\cos(\Phi_y)+\mathbf{I}\sin(\Phi_y)\right]\\ &\quad -2\mu\frac{\Delta x}{\Delta y}\left[1-\cos(\Phi_y)\right]\,. \end{aligned} \tag{4.27}$$

4.2.3 Fourier Symbol of Source Term

Last but not least the source is included in the analysis. As the source term is not spatially discretized the corresponding discretization operator $Z_S\circ$ given in Eq. (3.46) reads

$$Z_S\circ \;= \mathcal{V}\cdot S\cdot \mathcal{I}\circ \tag{4.28}$$

under the assumptions of Sect. 4.1, i. e. for the linear source term in Eq. (4.1). Applying Eq. (4.15) to $Z_S\circ$ instead of $Z_c\circ$ yields

$$\mathcal{Z}_S = \mathcal{V}\cdot S \tag{4.29}$$

as the Fourier symbol of the spatial discretization of the source term.

Merging the Fourier symbol of the semi–discretized model problem gives

$$\begin{aligned}\mathcal{Z}(\Phi_x,\Phi_y) &= \mathcal{Z}_c(\Phi_x,\Phi_y)+\mathcal{Z}_v(\Phi_x,\Phi_y)+\mathcal{Z}_S(\Phi_x,\Phi_y)\\ &= -u\Delta y\left[1-\cos(\Phi_x)+\mathbf{I}\sin(\Phi_x)\right]-v\Delta x\left[1-\cos(\Phi_y)+\mathbf{I}\sin(\Phi_y)\right]\\ &\quad -2\mu\frac{\Delta x}{\Delta y}\left[1-\cos(\Phi_y)\right]+\mathcal{V}S\,. \end{aligned} \tag{4.30}$$

4.3 Amplification Factor of the Time Stepping Scheme

The amplification factor g derived in this section represents the change of the amplitudes of the Fourier modes over the time of one iteration step, i. e.

$$\widehat{W}^{n+1} = g\cdot\widehat{W}^{n}\,. \tag{4.31}$$

g depends on the phase angles Φ_x and Φ_y just as on the flow parameters like e. g. the convection velocities. Furthermore the amplification factor depends on parameters according to the time stepping scheme and acceleration techniques. Since it affects only the amplitudes of the Fourier modes, the phase of the complex amplification factor is neglected when visualizing, i. e. $|g|$ is plotted instead.

In this section the time stepping scheme for the turbulence equations is analyzed. Using the operator notations for the time stepping scheme introduced in Sect. 3.2 leads to compact formulae for this analysis.
In Sect. 4.3.1 the amplification factor of the standard Runge–Kutta scheme is derived in analogy to [63]. Afterwards the amplification factor is modified concerning the point

implicit treatment of the turbulence source term as it is implemented in the FLOWer code. This is done in Sect. 4.3.2.

Although limitation of turbulence variables is performed on each Runge–Kutta stage it is not directly modeled within the framework of this local mode analysis: A lower bound is applied to the turbulence quantities only if destruction would drive them below the limit. Thus the bounding of the turbulence equations — if applied — equals the reduction or even switch off of the destructive effect of the source terms. This yields a lower bound $S_{\lim}$ for the source term S,

$$S_{\lim} \leq S \leq 0\,, \tag{4.32}$$

as the effect of the lower bound of the turbulence equations. The effect of different values for the source term coefficient S on damping properties is discussed in Sect. 5.2.

4.3.1 Amplification Factor of the Standard Runge–Kutta Scheme

Calculating the amplification factor of a time stepping scheme means to apply the time stepping scheme to a Fourier representation of a semi–discretized model problem. Here it is to apply the Runge–Kutta scheme, Eq. (3.54), to the Fourier representation

$$\frac{\partial \widehat{W}}{\partial t} e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y} = Z \circ \widehat{W} e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y} \tag{4.33}$$

of the model problem Eq. (4.1) with the operators $Z_* \circ$ derived in Sect. 4.2. By this one gets

$$\widehat{W}^{n+1} e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y} = \mathcal{G}_{\mathrm{ex}}^{(q=5)} \circ \widehat{W}^n e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y} \tag{4.34}$$

instead of Eq. (3.59). $\mathcal{G}_{\mathrm{ex}}^{(q=5)}\circ$ is still defined by Eq. (3.60). Eliminating $e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y}$ yields — by using the Fourier symbols $\mathcal{Z}_*$ corresponding to the spatial discretization operators in the definitions of $\mathcal{G}_{\mathrm{ex}}^{(q=5)} \circ$ and $f_{\mathrm{ex}}^{(q=5)} \circ$ —

$$\widehat{W}^{n+1} = \widehat{W}^n + \tau f_{\mathrm{ex}}^{(q=5)} \left(\mathcal{Z}_c + \mathcal{Z}_v + \mathcal{Z}_S\right) \widehat{W}^n = \widehat{W}^n + f_{\mathrm{ex}}^{(q=5)}\, \tau \mathcal{Z} \widehat{W}^n \tag{4.35}$$

with $f_{\mathrm{ex}}^{(q=5)}$ defined by

$$f_{\mathrm{ex}}^{(q)} = \alpha_q + \alpha_q \alpha_{q-1} \tau \mathcal{Z}_c + \ldots + \prod_{r=1}^{q} \alpha_r \left(\tau \mathcal{Z}_c\right)^{q-1}\,. \tag{4.36}$$

which is the same formula as in Eq. (3.57) except for the operators being replaced by their Fourier symbols.
Applying Eq. (4.31) to Eq. (4.35) yields the amplification factor

$$g_{\mathrm{ex}}^{(q=5)} = \frac{\widehat{W}^{n+1}}{\widehat{W}^n} = 1 + f_{\mathrm{ex}}^{(q=5)}\, \tau \mathcal{Z}\,. \tag{4.37}$$

Likewise in Sect. 3.2.1 the amplification factor of this Runge–Kutta scheme takes into account neither the point implicit treatment of the source term nor the implicit residual smoothing. The latter will be introduced in Sect. 4.4.1 whereas the effect of the point implicit scheme on the formula of the amplification factor g is discussed in the following section.

4.3.2 Amplification Factor of the Runge–Kutta Scheme with Point Implicit Source Term Treatment

Substituting $f_{\mathrm{ex}}^{(q=5)}$ in Eq. (4.37), with $f_{\mathrm{pi-src}}^{(q=5)}$ being modified in the same way as $f_{\mathrm{ex}}^{(q=5)} \circ$ from Eq. (3.57) to Eq. (4.36), $f_{\mathrm{pi-src}}^{(q=5)} \circ$, one derives

$$g_{\mathrm{pi-src}}^{(q=5)} = \frac{\widehat{W}^{n+1}}{\widehat{W}^{n}} = 1 + \tau f_{\mathrm{pi-src}}^{(q=5)} \mathcal{Z} \tag{4.38}$$

as the amplification factor of the Runge–Kutta time stepping scheme with point implicit treatment of the source term presented in Sect. 3.2.2, Eq. (3.68). $f_{\mathrm{pi-src}}^{(q)}$ reads

$$\begin{aligned} f_{\mathrm{pi-src}}^{(q)} &= \alpha_q \frac{1}{1-\alpha_q \tau \mathcal{Z}_S} \\ &\quad + \alpha_q \alpha_{q-1} \frac{1}{1-\alpha_q \tau \mathcal{Z}_S} \frac{1}{1-\alpha_{q-1} \tau \mathcal{Z}_S} \tau \mathcal{Z}_c \\ &\quad + \ldots + \prod_{r=1}^{q} \alpha_r \frac{1}{1-\alpha_r \tau \mathcal{Z}_S} (\tau \mathcal{Z}_c)^{q-1} \end{aligned} \tag{4.39}$$

because $\partial\left(Z_S \circ W^{(0)}\right)/\partial W$ reduces to $\mathcal{V} \cdot S \ (= \mathcal{Z}_S)$ for the linear source term of the model problem equation, Eq. (4.1).

According to the approximation of the source term Jacobian in Sect. 3.2.2.3 only negative source term contributions are treated implicitly. This approximation is captured by using $\min(0, \mathcal{Z}_S)$ instead of $\mathcal{Z}_S$ in the denominators occurring in the definition of $f_{\mathrm{pi-src}}^{(q)}$:

$$\begin{aligned} f_{\mathrm{pi-src}}^{(q)} &= \alpha_q \frac{1}{1-\alpha_q \tau \min(0, \mathcal{Z}_S)} \\ &\quad + \alpha_q \alpha_{q-1} \frac{1}{1-\alpha_q \tau \min(0, \mathcal{Z}_S)} \frac{1}{1-\alpha_{q-1} \tau \min(0, \mathcal{Z}_S)} \tau \mathcal{Z}_c \\ &\quad + \ldots + \prod_{r=1}^{q} \alpha_r \frac{1}{1-\alpha_r \tau \min(0, \mathcal{Z}_S)} (\tau \mathcal{Z}_c)^{q-1} \ . \end{aligned} \tag{4.40}$$

4.4 Fourier Analysis of Speed–Up Techniques

As described in Sect. 3.3 the numerical method implemented in the FLOWer code is accelerated by implicit residual smoothing and multigrid methods. The corresponding Fourier symbols are derived in the following sections 4.4.1 and 4.4.2 respectively. Also their influence on the amplification factor is discussed below.

Application of local time stepping has no effect on the amplification factor within this analysis as all control volumes for the model problem equation, Eq. (4.1), are of the same size because of equidistant point distributions. This together with identical velocities and viscosity everywhere in the grid yields that all local time steps are of the

same size, too. Due to the assumptions made for the stability analysis the time step defined in Eq. (3.73) can be simplified:

$$\Delta t_{i,j} = \frac{\mathcal{V}_{i,j}^2}{\mathcal{V}_{i,j} \cdot (u\Delta y + v\Delta x) + \texttt{DTVI} \cdot \mu \cdot (\Delta x^2 + \Delta y^2)} \,. \tag{4.41}$$

4.4.1 Fourier Symbol of Implicit Residual Smoothing

In this section the Fourier symbol of the implicit residual smoothing as introduced in Sect. 3.3.2 will be derived as in [63]. Also the corresponding changes to the formula of the amplification factor when applying this smoothing are presented.

Applying the implicit smoothing operator (in i–direction, Eq. (3.79)) to an arbitrary mode $\widehat{W}e^{\mathbf{I}i\Phi_x+\mathbf{I}j\Phi_y}$ reads

$$B_i \circ \mathcal{B}_i \cdot \widehat{W}e^{\mathbf{I}i\Phi_x+\mathbf{I}j\Phi_y} = \widehat{W}e^{\mathbf{I}i\Phi_x+\mathbf{I}j\Phi_y} \,. \tag{4.42}$$

$\mathcal{B}_i$ stands for the effect of this smoothing on the amplitude $\widehat{W}$, i. e. $\mathcal{B}_i$ is the Fourier symbol of the inverted implicit smoothing operator $B_i^{-1} \circ$. In Eq. (4.42) $\widehat{W}e^{\mathbf{I}i\Phi_x+\mathbf{I}j\Phi_y}$ corresponds to the qth stage residual $R_{i,j}^{(q-1)}$ in Eq. (3.77a) whereas $\mathcal{B}_i \cdot \widehat{W}e^{\mathbf{I}i\Phi_x+\mathbf{I}j\Phi_y}$ corresponds to $R_{i,j}^{\dagger}$ in the same equation.

Inserting Eq. (3.79) to Eq. (4.42) and resolving it for $\mathcal{B}_i$ leads to

$$\mathcal{B}_i = \frac{1}{1 + \varepsilon_i \left(2 - e^{\mathbf{I}\Phi_x} - e^{-\mathbf{I}\Phi_x}\right)} \,. \tag{4.43}$$

Using trigonometric functions this simplifies to

$$\mathcal{B}_i = \frac{1}{1 + 2\varepsilon_i \left(1 - \cos(\Phi_x)\right)} \tag{4.44}$$

which is the Fourier symbol of implicit residual smoothing in i–direction. Analogously one gets

$$\mathcal{B}_j = \frac{1}{1 + 2\varepsilon_j \left(1 - \cos(\Phi_y)\right)} \tag{4.45}$$

for the corresponding symbol of the j–direction. As the coordinate directions are treated one after the other the Fourier symbol of the whole implicit residual smoothing is reached by multiplying the coordinate dependent symbols:

$$\mathcal{B} = \mathcal{B}_j \mathcal{B}_i \,. \tag{4.46}$$

The implicit residual smoothing Fourier symbol of course depends on the smoothing coefficients:

$$\mathcal{B} = \mathcal{B}(\varepsilon_i, \varepsilon_j) \tag{4.47}$$

in the same way as the corresponding operator, Eq. (3.82).

Applying the implicit smoothing to each stage residual yields a modified characterizing polynomial $f^{(q)}_{\mathrm{pi-src,\,IRS}}$ of the qth Runge–Kutta stage acting on the Fourier symbols pursuant to

$$\begin{aligned} f^{(q)}_{\mathrm{pi-src,\,IRS}} \;=\; \mathcal{B}\Bigg[& \frac{\alpha_q}{1-\alpha_q\tau\min\left(0,\mathcal{Z}_S\right)} \\ & + \frac{\alpha_q}{1-\alpha_q\tau\min\left(0,\mathcal{Z}_S\right)}\,\frac{\alpha_{q-1}}{1-\alpha_{q-1}\tau\min\left(0,\mathcal{Z}_S\right)}\,\tau\mathcal{Z}_c\mathcal{B} \\ & + \ldots + \prod_{r=1}^{q}\frac{\alpha_r}{1-\alpha_r\tau\min\left(0,\mathcal{Z}_S\right)}\left(\tau\mathcal{Z}_c\mathcal{B}\right)^{q-1}\Bigg]\,. \end{aligned} \tag{4.48}$$

By use of this modified polynomial the Runge–Kutta scheme acting on the amplitude $\widehat{W}$ with point implicit source term treatment and implicit residual smoothing can be written as

$$\widehat{W}^{n+1} = \widehat{W}^{n} + \tau f^{(q=5)}_{\mathrm{pi-src,\,IRS}}\,\mathcal{Z}\widehat{W}^{n}\,. \tag{4.49}$$

From Eq. (4.49) the modified amplification factor g of the Runge–Kutta time stepping scheme can be directly derived as

$$g^{(q=5)}_{\mathrm{pi-src,\,IRS}} = \frac{\widehat{W}^{n+1}}{\widehat{W}^{n}} = 1 + \tau f^{(q=5)}_{\mathrm{pi-src,\,IRS}}\,\mathcal{Z}\,. \tag{4.50}$$

4.4.2 Amplification Factor of the Multigrid Method

The multigrid method does not influence the Runge–Kutta scheme itself but modifies the amplification factor of the total time stepping scheme for which $g^{(q=5)}_{\mathrm{ex}}$ equals a single grid amplification factor as mentioned in Sect. 3.3.3.1.
The amplification factor has been derived by Blazek [6] only for a two level scheme. This section enables to investigate an arbitrary number of multigrid levels. However, Jameson [51] presented corresponding results from a Fourier analysis of the multigrid speed–up for the Euler equations using up to six multigrid levels. The derivation given here includes viscous fluxes as well as source terms in extension to the one of Jameson.

As already noted in Sect. 4.3 the procedure ensuring positivity of the turbulence quantities is not explicitly modeled by this local mode analysis. Therefore it is not considered for the amplification factor of the multigrid method either.

4.4.2.1 Amplification Factor of Fine Grid Operator

From Eq. (4.49) within the previous section it is known that the amplitude after one Runge–Kutta step on one grid level is given by

$$\widehat{W}^{n+1}_{h,\,\mathrm{SG}} = \widehat{W}^{n}_{h} + \tau_h f^{(q=5)}_{\mathrm{pi-src,\,IRS},\,h}\,\mathcal{Z}_h\widehat{W}^{n}_{h} \tag{4.51}$$

with (compare Eq. (3.86) resp. Eq. (4.35))

$$\mathcal{Z} = \mathcal{Z}_c + \mathcal{Z}_v + \mathcal{Z}_S\,. \tag{4.52}$$

The supplementary subscript h again indicates fine grid values. As already described in Sect. 3.3.3 the fine grid residuals are implicitly smoothed with a constant coefficient before being transferred to the coarse grid. Therefore the amplitude $\widehat{W}^{n+1}_{h,\mathrm{SG}}$ changes to

$$\widehat{W}^{n+1}_{h,\mathrm{FG}} = \widehat{W}^{n}_{h} + \mathcal{B}_{\mathrm{MG}}\tau_h f^{(q=5)}_{\mathrm{pi-src,\,IRS},\,h}\,\mathcal{Z}_h\widehat{W}^{n}_{h}\,. \tag{4.53}$$

which is the amplitude after one Runge–Kutta cycle on the fine grid within a multigrid cycle. Here $\mathcal{B}_{\mathrm{MG}}$ is the same Fourier symbol as $\mathcal{B}_h$ but depends on the constant coefficients $(\varepsilon_{\mathrm{MG}}, \varepsilon_{\mathrm{MG}})$ instead of the variable coefficients $(\varepsilon_i, \varepsilon_j)$.

From Eq. (4.53) it can be easily derived that the amplification factor corresponding to $\mathcal{G}^{(q=5),\,\mathrm{FG}}_{\mathrm{pi-src,\,IRS},\,h}\circ$ as given in Eq. (3.88) reads

$$g^{(q=5),\,\mathrm{FG}}_{\mathrm{pi-src,\,IRS}} = 1 + \mathcal{B}_{\mathrm{MG}}\tau_h f^{(q=5)}_{\mathrm{pi-src,\,IRS},\,h}\,\mathcal{Z}_h\,. \tag{4.54}$$

4.4.2.2 Fourier Symbol of Restriction Operator

Two cases have to be distinguished for the data transfer from a fine to a coarse grid (cf. Sect. 3.3.3): The variables are injected from the fine to the coarse grid whereas the residuals needed for the forcing function are restricted to the coarse grid by a volume weighting. As the values on the fine and on the coarse grid are identical for coinciding points the Fourier symbol corresponding to $\widetilde{I}^{2h}_{h}\circ$ is equal to 1, thus this symbol can be neglected.

The restriction operator for the residuals reads in one dimension

$$I^{2h}_{h}\Big|^{i}\circ\, R^{n+1}_{h,\,2i,2j} = \frac{1}{2}R^{n+1}_{h,\,2i-1,2j} + R^{n+1}_{h,\,2i,2j} + \frac{1}{2}R^{n+1}_{h,\,2i+1,2j}\,. \tag{3.91}$$

Here the assumption of identical volumes (Sect. 3.3.3.2) holds due to the equidistant point distributions this analysis is based on.

The restriction operator $I^{2h}_{h}\circ$ can be written as two one dimensional restriction operators (cf. Sect. 3.3.3 and Blazek [6]):

$$I^{2h}_{h}\circ \;=\; I^{2h}_{h}\Big|^{j}\circ\; I^{2h}_{h}\Big|^{i}\circ\;. \tag{3.95}$$

Therefore the Fourier symbol of the two dimensional restriction equals the product of the Fourier symbols of the one dimensional restriction operators. Applying $I^{2h}_{h}\Big|^{i}\circ$, Eq. (3.93), to $\widehat{W}e^{\mathbf{I}2i\Phi_x+\mathbf{I}2j\Phi_y}$ yields

$$\begin{aligned} &I^{2h}_{h}\Big|^{i}\circ\,\widehat{W}e^{\mathbf{I}2i\Phi_x+\mathbf{I}2j\Phi_y} \\ &\quad= \quad \mathcal{R}^{2h}_{h}\Big|_{i}\cdot\widehat{W}e^{\mathbf{I}2i\Phi_x+\mathbf{I}2j\Phi_y} \\ &\quad= \quad \frac{1}{2}\widehat{W}e^{\mathbf{I}(2i-1)\Phi_x+\mathbf{I}2j\Phi_y} + \widehat{W}e^{\mathbf{I}2i\Phi_x+\mathbf{I}2j\Phi_y} + \frac{1}{2}\widehat{W}e^{\mathbf{I}(2i+1)\Phi_x+\mathbf{I}2j\Phi_y}\,. \end{aligned} \tag{4.55}$$

Resolving this for the Fourier symbol $\mathcal{R}^{2h}_{h}\Big|_{i}$ of the one dimensional restriction operator in i–direction results in

$$\mathcal{R}^{2h}_{h}\Big|_{i} = \frac{1}{2}e^{-\mathbf{I}\Phi_x} + 1 + \frac{1}{2}e^{\mathbf{I}\Phi_x} = 1 + \cos(\Phi_x)\,. \tag{4.56}$$

Analogously the Fourier symbol for the second dimension reads

$$\mathcal{R}_h^{2h}\Big|_j = \frac{1}{2}e^{-\mathbf{I}\Phi_y} + 1 + \frac{1}{2}e^{\mathbf{I}\Phi_y} = 1 + \cos(\Phi_y)\,. \tag{4.57}$$

The total Fourier symbol of the two dimensional restriction operator is given by

$$\mathcal{R} = \mathcal{R}_h^{2h}\Big|_j\, \mathcal{R}_h^{2h}\Big|_i = \left[1 + \cos(\Phi_y)\right]\left[1 + \cos(\Phi_x)\right]\,. \tag{4.58}$$

4.4.2.3 Amplification Factor of Coarse Grid Operator

In the same way by which the amplification factor g has been derived from Eq. (3.60) in Sect. 4.3 the amplification factor representing the influence of the Runge–Kutta time stepping on the coarse grid is derived from Eq. (3.105):

$$g_{\mathrm{pi-src,\,IRS,\,2h}}^{(q=5),\,\mathrm{CG}} = 1 + \tau_{2h} f_{\mathrm{pi-src,\,IRS,\,2h}}^{(q=5)}\,\left(\mathcal{Z}_{2h} + \mathcal{F}_{2h}\right)\,. \tag{4.59}$$

The Fourier symbol $\mathcal{F}_{2h}$ of the forcing function operator Eq. (3.104) is given by

$$\mathcal{F}_{2H} = \left[\mathcal{R}_H^{2H}\mathcal{Z}_H - \mathcal{Z}_{2H}\right] g_{\mathrm{pi-src,\,IRS,\,}H}^{(q=5),\,\mathrm{CG}} + \mathcal{R}_H^{2H}\mathcal{F}_H \tag{4.60}$$

which is the Fourier transform of Eq. (3.121). Again $H \overset{\mathrm{def}}{=} 2^{\mathtt{level}-1}h$ for any multigrid level `level` is used.

4.4.2.4 Fourier Symbol of Prolongation Operator

From Sect. 3.3.3.4 it is known that three cases have to be distinguished for the transfer of coarse grid corrections on a two dimensional grid. Thus three different Fourier symbols are derived: As no modification of the coarse grid value takes place when performing the injection to the fine grid (cf. list item no. 1 and Eq. (3.107) with $c_{i,j} = 1$) the Fourier symbol $\widetilde{\mathcal{P}}_{2h}^{h}$ of the injection operator $\widetilde{I}_{2h}^{h}\circ$ equals

$$\widetilde{\mathcal{P}}_{2h}^{h} = 1\,. \tag{4.61}$$

For fine grid points according to list item no. 2 a single interpolation in i–direction (or j–direction) is performed based on coarse grid corrections already injected:

$$I_{[2]h}^{h}\Big|^i \circ \Delta W_{h,\,2i+1,2j} = a^{-}\Delta W_{h,\,2i,2j} + a^{+}\Delta W_{h,\,2i+2,2j}\,. \tag{3.109}$$

Inserting a Fourier representation of the coarse grid correction to this equation yields

$$\mathcal{P}_{[2]h}^{h}\Big|_i \cdot \widehat{W} e^{\mathbf{I}(2i+1)\Phi_x + \mathbf{I}2j\Phi_y} = a^{-}\widehat{W} e^{\mathbf{I}2i\Phi_x + \mathbf{I}2j\Phi_y} + a^{+}\widehat{W} e^{\mathbf{I}(2i+2)\Phi_x + \mathbf{I}2j\Phi_y}\,. \tag{4.62}$$

Resolving this for the Fourier symbol $\mathcal{P}_{[2]h}^{h}\Big|_i$ leads to

$$\begin{aligned} \mathcal{P}_{[2]h}^{h}\Big|_i &= a^{-}e^{-\mathbf{I}\Phi_x} + a^{+}e^{\mathbf{I}\Phi_x} \\ &= a^{-}\left[\cos\left(\Phi_x\right) - \mathbf{I}\sin\left(\Phi_x\right)\right] + a^{+}\left[\cos\left(\Phi_x\right) + \mathbf{I}\sin\left(\Phi_x\right)\right]\,. \end{aligned} \tag{4.63}$$

Analogously the Fourier symbol of the linear interpolation in j–direction can be derived as

$$\begin{aligned} \mathcal{P}^h_{[2]h}\Big|_j &= b^- e^{-\mathbf{I}\Phi_y} + b^+ e^{\mathbf{I}\Phi_y} \\ &= b^- \left[\cos\left(\Phi_y\right) - \mathbf{I}\sin\left(\Phi_y\right)\right] + b^+ \left[\cos\left(\Phi_y\right) + \mathbf{I}\sin\left(\Phi_y\right)\right] . \end{aligned} \tag{4.64}$$

For the fine grid points which are defined by list item no. 3 the Fourier symbol of the prolongation operator equals the product of the two one dimensional linear interpolation operator Fourier symbols and of the injection operator Fourier symbol:

$$\mathcal{P}^h_{2h} = \mathcal{P}^h_{[2]h}\Big|_j \, \mathcal{P}^h_{[2]h}\Big|_i \, \widetilde{\mathcal{P}}^h_{2h} \, . \tag{4.65}$$

4.4.2.5 Amplification Factor of the Two Grid Method

Rewriting Eq. (3.118) for the Fourier symbols yields the amplification factor of a two grid cycle as

$$g^{(q=5),\,2\mathrm{G}}_{\mathrm{pi-src,\,IRS}} = \left(1 + \mathcal{B}_{\mathrm{MG}} \mathcal{P}^h_{2h} \left[g^{(q=5),\,\mathrm{CG}}_{\mathrm{pi-src,\,IRS},\,2h} - 1\right]\right) g^{(q=5),\,\mathrm{FG}}_{\mathrm{pi-src,\,IRS},\,h} \tag{4.66}$$

with $g^{(q=5),\,\mathrm{CG}}_{\mathrm{pi-src,\,IRS},\,2h}$ defined by Eq. (4.59) and $g^{(q=5),\,\mathrm{FG}}_{\mathrm{pi-src,\,IRS},\,h}$ by Eq. (4.54).

4.4.2.6 Amplification Factor of the Multi Grid Method

The amplification factor of a multigrid method with an arbitrary number of grid levels is derived from Eq. (3.124) by Fourier transformation. This multigrid cycle amplification factor reads

$$\begin{aligned} g^{(q=5),\,\mathrm{MG}}_{\mathrm{pi-src,\,IRS}} = \Big(&1 + \mathcal{B}_{MG} \mathcal{P}^h_{2h} \left[g^{(q=5),\,\mathrm{CG}}_{\mathrm{pi-src,\,IRS},\,2h} - 1\right] \\ &+ \mathcal{B}_{MG} \mathcal{P}^h_{2h} \mathcal{P}^{2h}_{4h} \left[g^{(q=5),\,\mathrm{CG}}_{\mathrm{pi-src,\,IRS},\,4h} - g^{(q=5),\,\mathrm{CG}}_{\mathrm{pi-src,\,IRS},\,2h}\right] \\ &+ \ldots\Big) g^{(q=5),\,\mathrm{FG}}_{\mathrm{pi-src,\,IRS},\,h} \, . \end{aligned} \tag{4.67}$$

4.5 First Results

In this section first basic results of the Fourier analysis derived in this chapter are presented. Here the model problem is restricted to a simple convection–diffusion equation ($S = 0$) as the source term influence on the stability of the numerical scheme will be discussed in Chap. 5.

Since different discretizations are used for the convection terms of the RANS and the turbulence equations, the influence of the stage coefficients of the Runge–Kutta scheme is discussed in Sect. 4.5.1. Topic of Sect. 4.5.2 is the influence of the aspect ratio on the

damping properties of the Runge–Kutta scheme. The multigrid method improves the damping behaviour of the overall scheme as demonstrated in Sect. 4.5.3.

The basic settings for the analyses performed in this work are summarized here for convenience: A first order upwind discretization of the convective terms without splitting into a central and a dissipative part is considered. Thus numerical dissipation is taken into account on each Runge–Kutta stage. The viscous term is subject to a thin layer approximation in y–direction. It is evaluated on the first stage only. A 5–stage Runge–Kutta scheme is used with stage coefficients $\alpha = \left[\frac{1}{4}, \frac{1}{6}, \frac{3}{8}, \frac{1}{2}, 1\right]$ yielding a CFL number of 3.75. Implicit residual smoothing is applied on each stage with variable smoothing coefficients calculated according to [97]. Additionally they are limited to the interval $[0.2, 1.2]$. The influence of the cell aspect ratio on these smoothing coefficients is taken into account without any scaling, i. e. choose corresponding parameter `ZETA` in the FLOWer input equal to 0^*. It is assumed that the CFL number of the Runge–Kutta scheme without implicit smoothing can be increased by a factor of 2 (to 3) due to the smoothing [49, 54, 63]. The FLOWer parameter `DTVI` which scales the parabolic contribution to the time step Δt as in Eq. (3.73) is set to 4 based on experience [105].

In order to be able to compare the damping properties of the Runge–Kutta scheme through the following investigations the average over all Fourier modes of the absolute value of the amplification factor is defined as

$$\overline{\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right|}_{\mathrm{all}} = \frac{1}{(2N)^2} \sum_{(\Phi_x,\Phi_y)\in[-\pi,\pi]^2} \left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\ (\Phi_x,\Phi_y)\right| . \tag{4.68}$$

N — here $N = 90$ — is the number of points taken to resolve the computational domain of the model problem in both dimensions as used in Eq. (4.11). In the same way as defined in that equation (Φ_x, Φ_y) are taken as discrete quantities.
In analogy the mean value of $\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right|$ over the low error frequencies is defined:

$$\overline{\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right|}_{\mathrm{low}} = \frac{1}{N^2} \sum_{(\Phi_x,\Phi_y)\in[-\frac{\pi}{2},\frac{\pi}{2}]^2} \left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\ (\Phi_x,\Phi_y)\right| . \tag{4.69}$$

Both definitions change in case of multigrid:

$$\overline{\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right|}_{\mathrm{all}} = \frac{1}{(2N)^2} \sum_{(\Phi_x,\Phi_y)\in[-\pi,\pi]^2} \left|g^{(q=5),\,\mathrm{MG}}_{\mathrm{pi-src,\,IRS}}\ (\Phi_x,\Phi_y)\right| \tag{4.70}$$

and

$$\overline{\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right|}_{\mathrm{low}} = \frac{1}{N^2} \sum_{(\Phi_x,\Phi_y)\in[-\frac{\pi}{2},\frac{\pi}{2}]^2} \left|g^{(q=5),\,\mathrm{MG}}_{\mathrm{pi-src,\,IRS}}\ (\Phi_x,\Phi_y)\right| \tag{4.71}$$

are used if two or more multigrid levels are included to the analysis. However, these averages are named in the same way than their counterparts for singlegrid analysis as the correct definition will be obvious from the context the averages are used in.
Besides these averages the absolute value of the amplification factor at $(\Phi_x, \Phi_y) = (0, 0)$

*The FLOWer parameter `ZETA` is usually not chosen equal to 0 as the aspect ratio changes throughout the computational grid. Since the aspect ratio is a large constant for the Fourier analyses within this work `ZETA` $= 0$ is chosen. For the computational investigations for `ZETA` a value appropriate to the test case has been taken.

will be discussed if a source term is present. This value represents the effect of the numerical treatment of turbulence equations on very low–frequency modes. Their amplification factor approaches $\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\ (0,0)\right|$ for $(\Phi_x, \Phi_y) \to (0,0)$.
$\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\ (0,0)\right|$ gives additional information only source terms are present as in case $S = 0$

$$\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\ (0,0)\right| = 1 \tag{4.72}$$

is valid.
The same remarks according to multigrid as given for the averages over Fourier modes also hold for this specific value.

In Sect. 4.5.1 the influence of different sets of stage coefficients α_q on the damping properties of the Runge–Kutta method is investigated. Realistic settings not only for the aspect ratio of cell faces but also for the ratio of characteristic speeds and for the viscosity are presented in Sect. 4.5.2. The improvements of the damping properties of the numerical method introduced in Chap. 3 due to the application of multigrid are discussed in Sect. 4.5.3.

4.5.1 Influence of Runge–Kutta Coefficients on Amplification Factor

In this section the influence of the Runge–Kutta stage coefficients on the amplification factor of a single grid time stepping operator as given in Eq. (4.50) is discussed.

Van Leer et al. [71] investigated the damping properties of multi stage time stepping schemes for different spatial discretizations of the one dimensional Euler equations in relation to the stage coefficients used. They showed which stage coefficients have to be used for different spatial discretizations in order to reach optimal damping properties. Van Leer et al. restricted themselves to one dimension as "the extension of the analysis to the two–dimensional Euler equations is hampered by the lack of a model for characteristic time–stepping in two dimensions". Thus two dimensional effects have not been considered in the optimization of the Runge–Kutta stage coefficients. Also the influence of viscosity has not been considered. Furthermore the application of implicit smoothing was not included in the analysis.

The Runge–Kutta stage coefficients used in the FLOWer code are taken according to [71] to yield optimal damping properties for a central discretization of the convective terms: $\alpha_{\mathtt{cent}} = \left[\frac{1}{4}, \frac{1}{6}, \frac{3}{8}, \frac{1}{2}, 1\right]$. These coefficients are not optimized for the upwind discretization used for the convection of turbulence, i. e. $\alpha_{\mathtt{cent}}$ yield an unstable damping behaviour for an upwind discretized convection equation if no implicit residual smoothing is applied.

In Fig. 4.1 isolines of the absolute value of the amplification factor $\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right|$, Eq. (4.50), with Runge–Kutta coefficients optimized for a central discretization but using an upwind discretization for the convection terms are plotted. This combination is the standard setting for the numerical treatment of turbulence equations in the FLOWer code because only one set of stage coefficients is basically provided for the treatment of the whole system. It can be seen that the damping properties for the convection diffusion equation using implicit residual smoothing is stable.

This figure is to be compared to Fig. 4.2 which also shows isolines of the absolute value of the amplification factor $\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right|$ but using Runge–Kutta coefficients optimized for a first order upwind discretization, $\alpha_{\mathtt{upw}} = [0.0533, 0.1263, 0.2375, 0.4414, 1]$ [71]:
Although the isolines look the same in both figures the overall niveau of the amplification factor is higher for $\alpha_{\mathtt{upw}}$ than for $\alpha_{\mathtt{cent}}$: $\overline{\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right|}_{\mathrm{all}}(\alpha_{\mathtt{upw}}) \approx 0.693$ whereas $\overline{\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right|}_{\mathrm{all}}(\alpha_{\mathtt{cent}}) \approx 0.563$. One reason is the influence of viscosity: Reducing μ, i. e. tending to a convection dominated flow, exhibits the improved damping of the stage coefficients $\alpha_{\mathtt{upw}}$ in combination with the upwind discretization as it has been derived by van Leer et al. in their work [71]. This result is gained only if the implicit residual smoothing is decreased in addition.
Thus another reason for the better smoothing properties of the non–optimal stage coefficients seems to be the application of (central) implicit smoothing: Blazek [6] notes that "the central implicit smoothing operator in combination with an upwind discretization of the convective fluxes shows bad damping properties". He used $\alpha_{\mathtt{upw}}$ for his investigations.

Since the stage coefficients optimized for a central discretization, $\alpha_{\mathtt{cent}}$, are used in the FLOWer code even for the upwind discretized turbulence equations as default and test calculations did not show noteworthy differences in convergence behaviour these coefficients are used throughout the following investigations.

4.5.2 Influence of Aspect Ratio on Amplification Factor

In this section the influence of the aspect ratio of a grid cell on the amplification factor of a singlegrid time stepping operator as given in Eq. (4.50) is discussed.
While e. g. Zhu and Hirsch [134] consider only the cell face aspect ratio, here it is demonstrated that the ratio of the characteristic speeds is much more significant for the Fourier analysis of solution methods for convection equations at least. Furthermore, they use an aspect ratio of 100 for their Fourier analysis while their computations have to deal with an aspect ratio of $\mathcal{O}(10^3)$. Aspect ratio investigations have been done e. g. by Blazek [6] and Radespiel et al. [98], too. Both do not use aspect ratios larger than 10 for their investigations. Here the aspect ratio is taken to approximate the boundary layer area of a real numerical grid suitable for Navier–Stokes calculations.

In order to clarify the influence of the cell face aspect ratio on the damping properties as a first step $\mu = 0$ is chosen, i. e. only a simple convection equation is investigated: The amplification factor of the Runge–Kutta time stepping scheme with stage coefficients $\alpha_{\mathtt{cent}}$ combined with implicit residual smoothing applied to this convection equation is shown in Fig. 4.3. In Fig. 4.4 the same amplification factor is shown with the underlying grid having no longer an aspect ratio of 1 but of 5. The damping properties have been changed significantly due to the increase of the cell face aspect ratio.
Having a look at the Fourier symbol of the convection

$$\mathcal{Z}_c(\Phi_x, \Phi_y) = -u\Delta y\,[1 - \cos(\Phi_x) + \mathbf{I}\sin(\Phi_x)] - v\Delta x\,[1 - \cos(\Phi_y) + \mathbf{I}\sin(\Phi_y)] \tag{4.17}$$

no straight dependency on the cell face aspect ratios can be found. Thus it is not the aspect ratio which directly influences the damping properties but the variation in Δx

or Δy changes the convection's Fourier symbol, i. e. the characteristic speeds $v\Delta x$ and $u\Delta y$ are altered. The same effect influences the denominator of the time step, Eq. (4.41), thus the product of the time step per volume τ and the Fourier symbol $\mathcal{Z}_c$ is independent of the actual values of u, v, Δx and Δy if and only if the ratio of the characteristic speeds remains constant, e. g. $v\Delta x = u\Delta y$:

$$\begin{aligned}
\tau\mathcal{Z}_c &= \Delta t \frac{1}{\mathcal{V}_{i,j}}\mathcal{Z}_c \\
&= \mathtt{CFL}\,\frac{\mathcal{V}_{i,j}^2}{\mathcal{V}_{i,j}\left(u\Delta y + v\Delta x\right)}\frac{1}{\mathcal{V}_{i,j}}\left(-u\Delta y\left[1-e^{-\mathbf{I}\Phi_x}\right] - v\Delta x\left[1-e^{-\mathbf{I}\Phi_y}\right]\right) \\
&\overset{v\Delta x = u\Delta y}{=} \mathtt{CFL}\,\frac{\mathcal{V}_{i,j}}{2v\Delta x}\frac{1}{\mathcal{V}_{i,j}}\,v\Delta x\left(-\left[1-e^{-\mathbf{I}\Phi_x}\right] - \left[1-e^{-\mathbf{I}\Phi_y}\right]\right) \\
&= \mathtt{CFL}\,\frac{-\left[1-e^{-\mathbf{I}\Phi_x}\right] - \left[1-e^{-\mathbf{I}\Phi_y}\right]}{2}\,.
\end{aligned}$$

This result is verified in Fig. 4.5 where the convection speeds u and v have been chosen in such a way that the ratio of the characteristic speeds equals 1 while the cell face aspect ratio is 5 as in Fig. 4.4. Nevertheless the plot is the same as Fig. 4.3 — where $(v\Delta x)/(u\Delta y) = 1$, too.

It is important to keep in mind that the amplification factor is independent from the cell face aspect ratio only if no diffusion is taken into account, $\mu = 0$. In case of $\mu > 0$ the dependencies become more complicated since the Fourier symbol of the spatial discretization of the viscous term,

$$\mathcal{Z}_v(\Phi_x, \Phi_y) = -2\mu\frac{\Delta x}{\Delta y}\left[1-\cos(\Phi_y)\right]\,, \tag{4.26}$$

depends directly on the cell aspect ratio whereas the viscous contribution to the time step, Eq. (4.41), depends on the sum of the squared cell face areas. While it is — theoretically — possible to adapt a grid to the convection speeds in order to achieve an aspect ratio of characteristic speeds close to 1 throughout the whole computational domain one is not able to consider the viscosity coefficient μ at the same moment.
Thus — in addition to the cell face areas — the convection velocities u and v and the viscosity coefficient μ have to be specified as realistic as possible in order to gain usable results from the Fourier analysis: Within the boundary layer around a 2D RAE 2822 airfoil the numerical simulation of viscous flow (case 9 [25]) gave

$$u = 10^{-1}\,, v = 10^{-2}\,, \Delta x = 10^{-2}\,, \Delta y = 10^{-6}\,, \mu = 10^{-8} \tag{4.73}$$

as rough estimates for the required quantities which will be used as representative values in the following investigations. Since all of these values may vary significantly throughout the boundary layer Eq. (4.73) simply fixes one possible set of parameters. In Chap. 5 it will become clear that this restriction does not affect the problems due to the source term which are discussed there.
The damping properties of the time stepping scheme using $\alpha_{\mathtt{cent}}$ as stage coefficients and being applied to Eq. (4.1) with the parameters given in Eq. (4.73) is shown in Fig. 4.6: It is noteworthy that the very low frequent Fourier modes in y–direction are nearly undamped ($|g| > 0.9$) independent of the associated Fourier mode in x–direction. This is important since Fourier modes with a high frequency in one direction would experience a more efficient reduction even for low frequencies in the other

direction in case of lower characteristic speed ratio. For this compare the very efficient reduction of low frequent Fourier modes in x–direction associated with high frequent Fourier modes in y–direction.

4.5.3 Influence of Multigrid on Amplification Factor

The purpose of this section is to give some insight on the influence of the multigrid method on the overall amplification factor given in Eq. (4.67). Usually only two grid levels are considered for such an investigation [118, 114, 10]. Jameson [51] investigated the effect of multigrid with up to six multigrid levels for a 1D convection equation. Here a 2D convection–diffusion equation on a stretched grid is used. In general it is possible to use an arbitrary number of multigrid levels, however, due to the necessity to generate the grids typically not more than four levels are used within industrial applications.

Despite the fact that the implicit smoothing of multigrid corrections — introduced in Sect. 3.3.3.6 as $B_{\mathrm{MG}}^{-1}\circ$ in Eq. (3.124) — turned out to be crucial for convergence in applications, Fourier analysis does not show its benefits: Comparing e. g. Fig. 4.6 to Fig. 4.7 both Fourier footprints are based on the representative values presented in the previous section inserted into Eq. (4.54). The difference between both footprints is due to the value of the constant implicit smoothing coefficient $\varepsilon_{\mathrm{MG}}$which equals the parameter `EPSC` in FLOWer input. For Fig. 4.6 $\varepsilon_{\mathrm{MG}} = 0.0$ has been used while $\varepsilon_{\mathrm{MG}} = 0.2$ lead to the results given in Fig. 4.7. The damping properties are significantly reduced when this smoothing is applied: The average over all Fourier modes of $\overline{\left|g_{\mathrm{pi-src,\,IRS}}^{(q=5)}\right|}_{\mathrm{all}}(\varepsilon_{\mathrm{MG}} = 0.2) \approx 0.63$ whereas the mean value of $\overline{\left|g_{\mathrm{pi-src,\,IRS}}^{(q=5)}\right|}_{\mathrm{all}}(\varepsilon_{\mathrm{MG}} = 0.0) \approx 0.37$. Due to this negative effect of $B_{\mathrm{MG}}^{-1}\circ$ on the overall damping properties $\varepsilon_{\mathrm{MG}}$ is chosen equal to 0.0 for all following Fourier analyses. This goes along with Radespiel et al. [98]. They did not apply this smoothing for the Fourier analysis even though Radespiel suggested $\varepsilon_{\mathrm{MG}} = 0.2$ before in [94].

The improvement of the damping properties by the multigrid method is clearly demonstrated in Figs. 4.8 – 4.10. These are Fourier footprints corresponding to Fig. 4.6 but with two to four multigrid levels. As explained above the parameter $\varepsilon_{\mathrm{MG}}$ has been chosen equal to 0.0 for all these figures. As transfer operator for data from coarser to finer grids $I_{2h}^{h}\circ = \widetilde{I}_{2h}^{h}\circ$ as defined in Eq. (3.107), i. e. $\mathcal{P}_{2h}^{h} = 1$ has been used.
By using more levels the damping properties especially for the low error frequencies (within the dotted boxes) are significantly improved. Averages of the amplification factor over all Fourier modes $\overline{\left|g_{\mathrm{pi-src,\,IRS}}^{(q=5)}\right|}_{\mathrm{all}}$ and over all low frequencies $\overline{\left|g_{\mathrm{pi-src,\,IRS}}^{(q=5)}\right|}_{\mathrm{low}}$ respectively using different number of multigrid levels are listed in Tab. 4.1. These values clearly show the improved damping properties for the low Fourier modes: While $\overline{\left|g_{\mathrm{pi-src,\,IRS}}^{(q=5)}\right|}_{\mathrm{all}}$ reduces by less than 25% from one to four multigrid levels $\overline{\left|g_{\mathrm{pi-src,\,IRS}}^{(q=5)}\right|}_{\mathrm{low}}$ experiences a reduction of more than 40% on the same range of multigrid levels: With four multigrid levels being applied the low frequencies are damped out globally nearly at the same rate as all frequencies.

# multigrid level	$\overline{\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}} \right\rvert}_{\mathrm{all}}$	$\overline{\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}} \right\rvert}_{\mathrm{low}}$
1	0.37094	0.50231
2	0.30186	0.34866
3	0.29027	0.3051
4	0.2872	0.29282

Table 4.1: Amplification factor averages related to number of multigrid levels

4.6 Summary

In this chapter the Fourier symbols of all operators introduced in Chap. 3 have been deduced. These Fourier symbols have been used in the previous section to obtain some basic results of the Fourier analysis of a two dimensional convection–diffusion equation.
First the influence of the Runge–Kutta coefficients has been investigated. It has been found that coefficients optimized for a central discretization yield sufficiently good damping also for an upwind discretization provided implicit residual smoothing is applied.
Thereafter the characteristic speed ratio $(v\Delta x)/(u\Delta y)$ has been introduced in order to replace the geometric aspect ratio $\Delta x/\Delta y$ as the characteristic speed ratio is more meaningful with respect to the damping properties than the geometric aspect ratio.
At last the improvements of the damping behaviour due to the application of a multigrid approach using up to four multigrid levels have been demonstrated. In this context the smoothing of multigrid corrections has been neglected in the analysis, even though it was found to be crucial in applications.

From these results it can be concluded that the basic settings and algorithms of the FLOWer code yield satisfactory good damping properties for convection diffusion equations.

5 Influence of Turbulence Source Terms

Although state–of–the–art CFD applications nowadays are RANS calculations with advanced transport equation turbulence models — e. g. [81] — there is a great variety in treating the turbulence equations numerically [27, 127].

Turbulence models like the Wilcox k–ω model introduced in Sect. 2.2 have the same form as transport equations like the RANS equations but contain an additional source term. The influence of this source term on the numerical behaviour of the transport equation turbulence model is investigated in the following sections.

In Sect. 5.1 it will be shown that stability problems might arise from using transport equation turbulence models if multigrid is used to accelerate convergence for Reynolds numbers appropriate for flight simulation. By observations from several numerical calculations it becomes obvious that these difficulties arise from the (productive) source term within the turbulence equations. This numerical behaviour is investigated further in Sect. 5.2 by means of the Fourier analysis introduced in Chap. 4. Here a special emphasis is laid on the value of the source term in comparison to the eigenvalues of the spatial discretization operator applied to the turbulence equations. Sect. 5.3 closes this chapter by a short summary and some conclusions.

5.1 Numerical Observations

The starting point for this work was the objective to perform Navier–Stokes calculations at flight Reynolds numbers in a robust and efficient way. As a first step a viscous flow around a 2D RAE 2822 airfoil has been simulated at several Reynolds numbers using the algebraic turbulence model of Baldwin and Lomax [4]. Based on these results the grids for all Reynolds numbers have been improved by adapting the grid to an approximated boundary layer as suggested by Stock et al. [113]. The grid generation and adaptation is described in more detail in Sect. 5.1.1 while results for these Reynolds number adapted grids using the Baldwin/Lomax turbulence model are presented in Sect. 5.1.2. The same grids have been used for Navier–Stokes calculations applying the k–ω model by Wilcox [131, 132], too. The experiences with this 2–equation model, especially in combination with the application of the multigrid method, are described in Sect. 5.1.3. These calculations show that the main obstacles for performing viscous calculations at flight Reynolds numbers come from employing transport equation turbulence models with a source term. Further investigations on this point are made

in Sect. 5.1.4 for turbulence decay which is an equilibrium between convection and destruction of turbulence quantities. Here it is demonstrated that not a source term in general is problematic but obviously one has to make a distinction between destructive and productive source terms. The section is concluded with a summary in Sect. 5.1.5.

An important quantity used throughout the following sections is the "residual", i. e. the averaged squared difference of a variable value from two consecutive time steps, which is related to Eq. (3.50). The residual, used subsequently as a measure for convergence, is calculated by the L_2 norm over the differences at all grid points, i. e.

$$\|\Delta W\|_2 = \sqrt{\frac{1}{N_i N_j} \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} \left(W_{i,j}^{n+1} - W_{i,j}^{n} \right)^2}. \tag{5.1}$$

Three different residuals are taken for convergence investigations: First the density residual is to represent the convergence of the RANS equations. The two other residuals are calculated from the turbulence variables k and ω if they are available.

5.1.1 Grid Generation for High Reynolds Number Flows

In this subsection the grid generation procedure is described. For the grid generation MegaCads [17], the grid generation package of the MEGAFLOW software [66, 82], has been used.

Grid generation is an important task (not only) when looking for high or even flight Reynolds number calculations since the boundary layer needs to be resolved by the grid in an adequate way. A sufficient number of points has to be allocated, in order to get a discrete representation of gradients in wall normal direction within the boundary layer, fine enough to yield any information needed e. g. by the turbulence model in use. Within this work about thirty points are used for this objective. Because of the approximation of the boundary layer thickness as well as the elliptic smoothing applied to parts of the grids no fixed value can be given.
Beside the number of points used for discretization of the boundary layer a proper distribution of these points is necessary for an adequate boundary layer resolution, too. For this two values have to be adapted after choosing the number of points being distributed within the boundary layer: The boundary layer thickness itself and the first spacing in wall normal direction. The adaptation of the latter is described in Sect. 5.1.1.2 while the approximation of the boundary layer thickness is discussed in Sect. 5.1.1.1. The resulting quality of resolving the boundary layer is discussed in Sect. 5.1.1.3 instancing a velocity profile on the upper side of the RAE 2822 airfoil. In Sect. 5.1.1.4 some conclusions related to grid quality are drawn.

5.1.1.1 Approximation of Boundary Layer Thickness δ

The boundary layer thickness δ is approximated according to the work of Stock and Haase [113] which is based on the algebraic turbulence model developed by Baldwin and Lomax [4]. For the Baldwin/Lomax turbulence model the value $y_{\max}$ is calculated as the distance normal to the wall where the maximum of the product of the wall

distance and the absolute value of the vorticity occurs. By some analysis Stock et al. derive that $y_{\max}$ is proportional to the boundary layer thickness δ. They calibrated their approach consistently with boundary layer measurements for the RAE 2822 airfoil [25], test case no. 9, i. e. the flow conditions are $M_\infty = 0.73$, $\alpha = 2.8^\circ$ resp. Re $= 6.5 \cdot 10^6$. The same test case has been used as a baseline test case within this work.

Due to grid generation purposes the approximated boundary layer thickness δ can not be applied at every point on the profile but some special supporting points have been chosen: In the nose area three supports are located, the leading edge and the upper and lower transition points at 3% chord length each. On the upper side of the profile another support is located within the shock foot area at about 57% chord length as well as at the trailing edge. At the trailing edge on the lower side of the airfoil a support is placed as well. Two auxiliary supports on the lower side have been taken at about 31% and at about 72% chord length in order to maintain some level of control over the approximated boundary layer thickness between the supports on the lower side of the profile. All these supports are shown in the upper half of Fig. 5.1 together with a first — very crude — initial approximation of δ. This figure shows only a schematic zoom of the boundary layer part of a C–type grid around the RAE 2822 airfoil with a farfield distance of 20 chord lengths.
By solving the problem on a grid of this kind — via a Navier–Stokes calculation using the Baldwin/Lomax turbulence model — a better approximation of the boundary layer thickness can be achieved. This new δ–approximation has been used for the generation of a new grid with an improved boundary layer resolution. The lower half of Fig. 5.1 shows such an adapted boundary layer area for the RAE 2822 airfoil. After two more flow calculations which yielded further improved approximations of the boundary layer thickness, the generation of subsequently adapted grids did not show relevant changes anymore.
At farfield the wake thickness has been increased by a factor of 5 compared to its thickness (= δ) at the trailing edge. By widening the wake area more than by a factor of 5, e. g. by 20, the grid quality related to the stretching of cells can be improved further since the stretching of grid cells away from the body usually is not necessary but might inhibit convergence due to reduced damping properties of the Runge–Kutta time stepping scheme on stretched cells. Nevertheless this unexploited grid quality has been accepted for two reasons: One is the problem that in three dimensions the control of the grid quality becomes much more complicated, thus ranges of unnecessarily stretched cells have to be accepted as their correction might be too time consuming if possible at all. The second reason is that the problems discussed within this work were encountered even on grids with a more flared wake area.

This grid (Re $= 6.5 \cdot 10^6$) as sketched in Fig. 5.2 and grids for other Reynolds numbers — Re $= 20 \cdot 10^6$, Re $= 60 \cdot 10^6$ and Re $= 100 \cdot 10^6$ — generated by the same iterative procedure are used throughout this work. All these grids share the same point distribution on the profile (256 points) and along the wake (56 points) as well as the same number of grid points (88 points) in wall normal direction.

5.1.1.2 Adaptation of y^+

Additionally to the boundary layer thickness δ the first spacing in wall normal direction Δy_{w} is adapted during the same iterative procedure. A modified value for this

spacing at the same supports used for the approximation of δ is calculated from the first dimensionless distance in wall normal direction,

$$y_{\mathrm{w}}^{+} = \frac{\Delta y_{\mathrm{w}} \sqrt{\rho \tau_{\mathrm{w}}}}{\mu_l} \,, \tag{5.2}$$

such that y_{w}^{+} equals 1. It has been found that two adaptation steps yielded negligible changes in Δy_{w}, too.

Aiming at y_{w}^{+} close to 1 is based on the assumption that the viscous (also called laminar) sublayer within a boundary layer has a thickness of $y^{+} \leq 5$ as stated e. g. by Truckenbrodt [119]. Thus $y_{\mathrm{w}}^{+} \approx 1$ is supposed to achieve at least 3 grid points in wall normal direction within the viscous sublayer which is assumed to yield a sufficiently fine resolution [132].
In Fig. 5.3 the distribution of y_{w}^{+} around the RAE 2822 airfoil on the Re–adapted grids is shown. While y_{w}^{+} equals 1 at all supports and between the auxiliary supports (at about 31% and at about 72% chord length on the lower side) y_{w}^{+} is less than 2.5 for all Reynolds numbers. The abrupt increase of y_{w}^{+} just behind the supports at the transition points at 3% of chord length is directly related to an increase of the wall shear stress τ_{w} due to laminar–turbulent transition.

The adaptation of y_{w}^{+} leads to an increase of the cell face aspect ratio of the first cells adjacent to the wall. This increase scales with the Reynolds number as shown in Fig. 5.4.

5.1.1.3 Resolution of Velocity Profile

One possibility to check the resolution of the boundary layer is to have a look at the computed dimensionless velocity profile. For all Reynolds numbers under consideration the velocity profiles at 40% chord length on the upper side of the RAE 2822 airfoil have been calculated from fully converged k–ω singlegrid solutions. While the k–ω calculations are discussed later in Sect. 5.1.3 the velocity profiles are presented in Fig. 5.5 together with an analytical solution.

The analytical reference solution usually consists of three parts. One formula describes the dimensionless velocity profile in the laminar sublayer, a second one is related to the fully turbulent part of the boundary layer. The run of the velocity profile in the transitional region between these two parts of the boundary layer may be described with an additional formula. This formula has been skipped since there are several possibilities to choose [119].
Within the laminar sublayer the analytical reference is given by

$$u^{+} = y^{+} \qquad \text{for} \quad y^{+} \leq 10$$

while the solution of the logarithmic law of wall is taken as reference for larger values of y^{+}:

$$u^{+} = \frac{1}{\kappa} \ln y^{+} + B \qquad \text{for} \quad y^{+} > 10 \,.$$

Usually the parameters κ and B are experimentally determined. Here

$$\kappa = 0.374$$

has been chosen in accordance to Eisfeld et al. [34] who adapted this parameter to fit the parameters of the k–ω turbulence model in use. Slightly different values derived from experiments are given by Schlichting et al. [108] and Truckenbrodt [119].

$$B = 5.24$$

has been taken from [119].

For all Reynolds numbers the computed velocity profiles reproduce the analytic reference throughout the laminar sublayer ($y^+ \leq 5$, [119]) as shown in Fig. 5.5. The transition from the laminar sublayer to the fully turbulent part of the boundary layer lacks correspondence with the analytical solution due to some features of the k–ω model used as discussed by Rudnik [105]. Rudnik also points out that this underdeveloped transition leads to a wrong estimation of the slope of the logarithmic law of wall by the k–ω turbulence model. Although Rudnik discussed the velocity profiles for the flat plate the computed velocity profiles for the RAE 2822 airfoil at different Reynolds numbers show the same behaviour. Further investigations on the parameters of the k–ω turbulence model implementation in the FLOWer code have been performed by Eisfeld and Rudnik [34] recently. They deduce some dependency of the slope of the dimensionless velocity profile in the logarithmic area on the choice / variation of these turbulence parameters. The improvement of turbulence models and their parameters is not an objective of this work.

5.1.1.4 Conclusions

Computational grids for the RAE 2822 airfoil at several Reynolds numbers have been generated. The process of adapting the grids to the boundary layers has been described in the previous sections. Besides an approximation of the boundary layer thickness the dimensionless distance in wall normal direction of the first point has been adapted to fit well for the corresponding Reynolds number. Thus these grids, each of them adapted to its own Reynolds number, are called "Re–adapted grids" within this work. The quality of boundary layer resolution has been discussed for the velocity profile at 40% chord length on the upper side of the airfoil.

It has been found that the iterative adaptation process yielded stable results within three iterations. While the boundary layer thickness itself has not been discussed, the distribution of y_w^+ over the airfoil has been investigated for all Reynolds numbers. Further improvement of y_w^+ might be gained by more supports or even a better placement of the existing supports. Still the velocity profiles show a good agreement with the analytical reference solution taking into account the problems of the k–ω turbulence model to meet the logarithmic law of wall.

It can be concluded that the Re–adapted grids generated for this work are well suited for resolving the boundary layer in an adequate manner. Further improvements are possible but their deployment is not part of this work.

5.1.2 Baldwin/Lomax Calculations for High Reynolds Number Flows

In this section the first results obtained on the Reynolds number adapted grids introduced in the previous section are presented. Since the DLR flow solver FLOWer has

been validated extensively [65, 18, 106, 88, 99] the resulting flow fields are taken to be correct and therefore discussed only briefly. Thus the focus is laid on investigating the convergence behaviour of the calculations performed.

These investigations are divided into three parts: Firstly the influence of the number of multigrid levels on the convergence behaviour at $\mathrm{Re} = 6.5 \cdot 10^6$ is discussed in Sect. 5.1.2.1. Secondly in Sect. 5.1.2.2 the number of multigrid levels is kept fixed while the Reynolds number in union with the correspondingly adapted grids is varied. Thirdly the correctness of the implementation of the multigrid method is shown in Sect. 5.1.2.3. This section is ended in Sect. 5.1.2.4 by a brief summary.

5.1.2.1 Baldwin/Lomax Calculations at $\mathrm{Re} = 6.5 \cdot 10^6$ Using Different Numbers of Multigrid Levels

Fig. 5.6 shows the convergence of the density residual for the RAE 2822 airfoil at flow conditions of the experimental test case 9 [25] i. e. freestream Mach number $\mathrm{M}_\infty = 0.73$, angle of attack $\alpha = 2.8^\circ$ and Reynolds number $\mathrm{Re} = 6.5 \cdot 10^6$. In order to visualize the results of Tab. 4.1 in Sect. 4.5.3 the number of multigrid levels has been varied. Although not shown to its full extent the singlegrid computation converges but needs more than 30 000 iterations to reach machine zero. Thus a speedup factor referring to the number of iterations of about 8 is gained by applying a 4 level V–cycle multigrid method to the same problem. This speedup is significantly larger than predicted in Sect. 4.5.3. On the one hand this has to do with the question which Fourier modes are resolved on the finest grid used for these computations since the largest amplification factor

$$\max_{(\Phi_x,\Phi_y)\in[-\pi,\pi]^2\setminus(0,0)} \left| g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\ (\Phi_x, \Phi_y) \right|$$

— related to the lowest mode present — limits the total convergence speed. Moreover implicit smoothing is applied to the multigrid corrections which has been neglected in Fourier analysis. Another reason is that in Sect. 4.5.3 only one special grid cell has been investigated at one fixed flow condition setting whereas in this application grid cells and local flow conditions vary over a large range within the flow field as well as during the iteration process. Thus synergy effects due to these variations could not be taken into account in the Fourier analysis. From these reasons it has to be concluded that the results obtained from Fourier analysis give useful information according to trends but no absolute value can be obtained from it within the current framework. More rigorous estimations by Fourier analysis of multigrid have been done e. g. by Brandt [11].

W–cycles are known to be advantageous over V–cycles. While for two multigrid levels W– and V–cycles are identical, more iterations on the coarser grid levels are performed in case of three or more multigrid levels yielding an increased damping of low Fourier modes. Applying W–cycles to the same test case leads to higher speedup as shown in Fig. 5.7. Only about 5 % of the number of singlegrid iterations is needed to reach machine zero using a 4 level W–cycle. A speed up concerning cpu time is also achieved since this multigrid cycle takes about as much operations as three singlegrid iterations.

5.1.2.2 Baldwin/Lomax Calculations at Different Reynolds Numbers Using 4 Level W–cycle

As a next step multigrid calculations using a 4 level W–cycle have been performed on all four Reynolds number adapted grids. ¿From Fig. 5.8 it can be seen that increasing the Reynolds number in union with using correspondingly adapted grids slows down convergence but does not perturb it. Increasing the Reynolds number by a factor of 15 ($\mathrm{Re} = 100 \cdot 10^6$ instead of $\mathrm{Re} = 6.5 \cdot 10^6$) leads to about 2.5 times as many multigrid cycles (about $4\,300$ instead of $1\,700$) for convergence to machine zero.

The distributions of the pressure coefficient resulting from these calculations are shown in Fig. 5.9. The corresponding result from an Euler calculation is also shown in this figure since the Euler equations can be regarded as the limit of the RANS equations under $\mathrm{Re} \to \infty$.
The distributions of the pressure coefficient C_p presented in Fig. 5.9 show, that at higher Reynolds numbers the shock tends to the position calculated in the Euler case, which is quite what one expects. Furthermore, the shock becomes stronger with increasing Reynolds number. In addition the suction peak as well as the whole plateau in front of the shock on the suction side is slightly raised with an increase of the Reynolds number, towards the Euler solution. Analogous behaviour can be observed on the lower side of the airfoil. From these observations it can be deduced that the results of the RANS calculations at different Reynolds numbers are reasonable.

5.1.2.3 Multigrid Restart from Singlegrid Solution for Baldwin/Lomax Calculations at $\mathrm{Re} = 6.5 \cdot 10^6$

In order to ensure that the multigrid method works correctly the solution of a singlegrid calculation at $\mathrm{Re} = 6.5 \cdot 10^6$ has been selected as a restart using no longer singlegrid but a W–multigrid cycle. The solution to restart from has been taken after 40 000 iterations with density residual equal to machine zero: Consistent with multigrid theory — a residual equal to zero on the finest grid results in a modified coarse grid residual as defined in Eq. (3.96) equal to zero, too, thus no coarse grid corrections appear — the residual stays at machine zero independent of how many multigrid levels have been used as shown in Fig. 5.10.

5.1.2.4 Summary

In the previous section three points have been shown: First and foremost the general ability of simulating high Reynolds number flows by applying the DLR flow solver FLOWer to reasonably adapted 2D grids was demonstrated, i. e. no convergence perturbations at flight Reynolds numbers for RANS calculations using the algebraic turbulence model by Baldwin and Lomax [4] on Reynolds number adapted grids have been found.
Secondly it has been been shown that the multigrid method in the FLOWer code works correctly for the RANS equations. Thirdly it became obvious that the Fourier analysis introduced in Chap. 4 is able to show correct trends but not to give absolute values.

5.1.3 k–ω Calculations for High Reynolds Number Flows

Since no convergence problems occurred simulating high Reynolds number flows using the 0–equation turbulence model by Baldwin and Lomax [4] as reported in the last section, a more advanced turbulence model has been applied to the same test cases. In this section the convergence behaviour using the k–ω turbulence model by Wilcox [132] as presented in Sect. 2.2 is discussed.
The input parameters for the FLOWer code are the same as in Sect. 5.1.2, with the exception of the parameters directly related to the turbulence model. The Reynolds number adapted grids are identical.

The same investigations performed with the Baldwin/Lomax turbulence model have been done using the k–ω model. In Sect. 5.1.3.1 the influence of varying numbers of multigrid levels on the convergence is discussed whereas the variation of the Reynolds number and its influence on the convergence behaviour is presented in Sect. 5.1.3.2. The effect of restarting from a singlegrid solution using a multigrid cycle is shown in Sect. 5.1.3.3. Additionally to the investigations in Sect. 5.1.2, influence on the convergence behaviour from applying different lower bounds for the turbulence quantities k and ω is investigated. The results are summarized in Sect. 5.1.3.4.

5.1.3.1 k–ω Calculations at Re $= 6.5 \cdot 10^6$ Using Different Numbers of Multigrid Levels

Navier–Stokes calculations using the Wilcox k–ω model have been performed for the RAE 2822 airfoil at the flow conditions of the experimental test case 9 [25]. The convergence of the density residual for a varying number of multigrid levels — in correspondence to Fig. 5.7 — are plotted in Fig. 5.11. Similar to the singlegrid calculation using the Baldwin/Lomax model, the calculation using only the finest grid is converging to machine accuracy within about 30 000 iterations.
The density residual does not reach machine zero in case of using a W–cycle with three or four levels of multigrid. The density residual lacks three to five orders of magnitude of accuracy compared to those calculations taking into account one or two levels of multigrid only. Applying a V–cycle with three to four levels of multigrid yields an improved level of convergence, cmp. Fig. 5.12, but still no machine accuracy could be reached.

Having a closer look at the aerodynamic coefficients for lift and drag calculated with different numbers of multigrid levels as given in Tab. 5.1, one can note that the application of the multigrid method in combination with the k–ω turbulence model does not only inhibit the density residual from converging to machine zero but also changes the aerodynamic coefficients. These changes are far from being relevant in terms of aerodynamic design decisions — below 0.4 lift counts and 0.2 drag counts* — since e. g. the difference in these coefficients due to the turbulence model chosen is much larger: The lift resp. drag coefficient calculated using the Baldwin/Lomax model are given in Tab. 5.2. Still these differences reach the fourth significant decimal place which is larger than indicated by the different convergence levels of the density residual. A further brief discussion of these differences is given in Sect. 6.3.6.

*A lift count equals 10^{-3} while a drag count denotes 10^{-4}.

# multigrid level	lift	drag
1	0.798361	0.018146
2	0.798371	0.0181464
3	0.798426	0.0181488
4	0.798727	0.018162

Table 5.1: Aerodynamic coefficients for lift and drag related to number of multigrid levels within W–cycle using k–ω turbulence model; RAE 2822 airfoil at Re $= 6.5 \cdot 10^6$

# multigrid level	lift	drag
1 – 4	0.805034	0.0181512

Table 5.2: Aerodynamic coefficients for lift and drag related to number of multigrid levels within W–cycle using Baldwin/Lomax turbulence model; RAE 2822 airfoil at Re $= 6.5 \cdot 10^6$

In Fig. 5.13 and Fig. 5.14 the convergence of the residuals related to the turbulence equations, k resp. ω, are shown in dependence of the number of multigrid levels used. While both residuals converge using singlegrid only, they do not reach machine accuracy if two or more levels of multigrid are applied. The ω residual converges more than ten orders of magnitude for all numbers of multigrid levels, while the k residual stops convergence before a decrease of six orders of magnitude is reached for any multigrid cycle. The same behaviour occurs if a V–cycle is applied instead of a W–cycle as shown in Figs. 5.15 – 5.16. Furthermore the differences in the convergence level reached for the different types of multigrid cycles are less significant for the k and ω residuals compared to the influence of the type of multigrid cycle on the convergence level of the density residual as in Fig. 5.12.

5.1.3.2 k–ω Calculations at Different Reynolds Numbers Using 4 Level W–cycle

When looking at the convergence behaviour at Reynolds numbers larger than Re $= 6.5 \cdot 10^6$ applying a 4 level W–cycle one observes perturbed, non–converging density residuals at Re $= 60 \cdot 10^6$ and Re $= 100 \cdot 10^6$ while at Re $= 20 \cdot 10^6$ the density residual converges about nine orders of magnitude as shown in Fig. 5.17. Thus one order of magnitude is lost compared to the convergence at Re $= 6.5 \cdot 10^6$. Corresponding behaviour can be observed for the k residual in Fig. 5.18 as well as for the ω residual in Fig. 5.19.

Since the Baldwin/Lomax calculations presented in Sect. 5.1.2.2 showed no convergence problems at any Reynolds number using the 4 level W–cycle the question arises whether the k–ω turbulence model is suited for higher Reynolds numbers in general or not.
In the previous section the singlegrid computation at Re $= 6.5 \cdot 10^6$ was found to converge to machine zero for all residuals. Performing singlegrid calculations at all Reynolds numbers yields convergence of the density residual to machine zero in all cases. Again the residuals of the turbulence quantities k and ω present an analogous

behaviour and reach machine zero, too. For these results compare Figs. 5.20 – 5.22. It is important to point out that no hindrance concerning the convergence of the k–ω calculations appeared.

These results suggest a more extensive investigation of the influence of multigrid on the convergence at high Reynolds numbers. Therefore additional calculations at $\mathrm{Re} = 60 \cdot 10^6$ using different numbers of multigrid levels as well as different types of multigrid cycles have been performed.
In Fig. 5.23 one can observe the deterioration of convergence of the density residual by applying a more powerful multigrid cycle: While a 2 level cycle still converges down to 10^{-12} using more levels of multigrid does not yield convergence as long as a W–cycle is applied. Switching to a V–cycle gains convergence to about 10^{-9} for three and four multigrid levels as shown in Fig. 5.24. As the convergence plots for the k and the ω residuals do not give new insight to the problem, they are not included in this work.

5.1.3.3 Multigrid Restart from Singlegrid Solution for k–ω Calculations at $\mathrm{Re} = 6.5 \cdot 10^6$

Since convergence problems arise by applying the multigrid approach to RANS computations using the k–ω turbulence model, the investigations done in Sect. 5.1.2.3 on the behaviour of the residual after restarting from a singlegrid solution using a multigrid cycle have been repeated this time using the k–ω model. The main objective is to verify that the multigrid method as it is implemented in the FLOWer code behaves correctly even if it is applied to turbulence equations.

5.1.3.3.1 Default Settings

Restarting with a W–multigrid cycle from a singlegrid solution obtained after 40 000 iterations using the Wilcox k–ω turbulence model exhibits significant changes in the residuals as shown in Figs. 5.25 – 5.27: All residuals jump by 5 to 10 orders of magnitude independent of the number of levels used after restart before they start to converge again. They do reach the same level of convergence like in computations using the corresponding multigrid cycle starting at freestream conditions.
In Figs. 5.28 – 5.30 close–ups of the behaviour of the residuals during the first ten iterations after restarting are presented. Special attention has to be directed to the calculation restarting with singlegrid because the restart option of FLOWer has to ensure undisturbed continuation of a calculation to be restarted as long as no relevant input parameters have been changed.
On the first iteration after restart the density residual (Fig. 5.28) behaves as expected: As shown in Fig. 5.10 a restart from a singlegrid solution using singlegrid does not affect the density residual. The second iteration exhibits a severe change in the density residual, even for the restart applying only singlegrid. This perturbation is due to the jump of the residuals of k and ω in the first iteration after restart as it can be observed in Fig. 5.29 resp. Fig. 5.30. The reason for this perturbation of the k and the ω residual is the additional implicit smoothing (compare Sect. 3.3.3.1) within a multigrid cycle to which the fine grid residuals as well as the coarse grid corrections are subject to. The FLOWer input parameter EPSC — which corresponds to $\varepsilon_{\mathrm{MG}}$ in Sect. 3.3.3.1 — has been set to the singlegrid default 0.0 for the singlegrid solution restarted from.

After restart this parameter has been used with the multigrid default EPSC $= 0.2$ also for the continued singlegrid computation. This implicit smoothing with the constant coefficient EPSC $= 0.2$ is applied to residuals of the turbulence equations which might lead to negative values of k and ω respectively. Implicit smoothing of these unconstrained residuals perturbs the whole flow field which results in a difference between the turbulence variable values before and after the Runge–Kutta step which are used to calculate the residual.

5.1.3.3.2 Modified Singlegrid Settings

The singlegrid solution used for restart with multigrid, if the k–ω turbulence model is applied, has to be obtained using EPSC $= 0.2$ as FLOWer input instead of EPSC $= 0.0$ in order to ensure a correct transition of all residuals at restart. Close–ups of the convergence at restart are given in Figs. 5.31 – 5.33.
At first they show no change in any residual after restart using a singlegrid approach only. The k and ω residuals are not changed over the first iteration after restart, independent of the number of multigrid levels used for restart: Since the residuals are always calculated at the end of a Runge–Kutta step on the finest grid, the first residual after restart represents the first Runge–Kutta step on the finest grid and does not include any multigrid cycle.
It should be noted that all residuals show a perturbation in the second iteration, if more than one multigrid level is used. Since the RANS equations are treated correctly within the multigrid cycle as established in Sect. 5.1.2.3 these perturbations come from the multigrid treatment of the k–ω turbulence model equations.

Further results from the application of multigrid to the turbulence equations are given in the following section under particular consideration of the lower bound for the turbulence variables.

5.1.3.3.3 Modified Setting of Limiter for Turbulence Quantities

In this section the application of the multigrid method to the k–ω turbulence model is investigated with emphasis on the lower bound of the turbulence variables as presented in Sect. 3.2.3 within a multigrid cycle.

Basic Investigations
It has been found that the choice of the limiting factors $\lim_k$ and $\lim_\omega$ as introduced in Eq. (3.71) resp. Eq. (3.72) influences the convergence level of all residuals of multigrid computations using the k–ω model if some convergence level could be reached using the default values 10^{-5}. The FLOWer input parameter representing these limiting factors is RTULF. Although FLOWer reads two independent values for RTULF it has not been investigated whether different values for $\lim_k$ and $\lim_\omega$ could yield improved results or not.

In Figs. 5.34 – 5.36 the effect on the level of convergence reached at Re $= 6.5 \cdot 10^6$ using a 4 level W–cycle is shown for decreasing values of RTULF. The density residual as well as the ω residual converge to machine accuracy for RTULF $\leq 10^{-6}$ as shown in Fig. 5.34 resp. Fig. 5.36. The convergence of the k residual experiences significant improvement

in Fig. 5.35, but still does not converge to machine accuracy till RTULF $\leq 10^{-10}$.
On the other hand RTULF can not be reduced arbitrarily: At Re $= 20 \cdot 10^6$ convergence of the density residual to machine accuracy is achieved if RTULF is reduced to 10^{-7}. Further reduction of RTULF to 10^{-8} leads to a destabilized calculation i. e. no results could be obtained because negative values for pressure and density have been computed which are nonphysical in this context. For RTULF $= 10^{-9}$ again the calculation converges to machine accuracy with respect to the density residual. This result is presented in Fig. 5.37. Once more the convergence behaviour of the ω equation appears to be the same like for the density residual as shown in Fig. 5.39. This figure shows more clearly the stability problems which might arise from reducing RTULF: For RTULF $= 10^{-7}$ the convergence shows significant disturbances on the first thousand iterations. From Fig. 5.38 it can be seen that again the k–residual reaches machine accuracy only for RTULF $= 10^{-10}$. In contrast to the situation at Re $= 6.5 \cdot 10^6$ further reduction of RTULF leads to a destabilized computation, too.
These results show that there is no general setting for RTULF to ensure convergence to machine accuracy on the one hand and to keep stability on the other hand.

# multigrid level	lift	drag
1 – 4	0.798954	0.0181736

Table 5.3: Aerodynamic coefficients for lift and drag related to number of multigrid levels within W–cycle using k–ω turbulence model using RTULF $= 10^{-10}$; RAE 2822 airfoil at Re $= 6.5 \cdot 10^6$

Comparing the aerodynamic coefficients for lift and drag calculated at Re $= 6.5 \cdot 10^6$ with different numbers of multigrid levels using RTULF $= 10^{-10}$ as given in Tab. 5.3 with those given in Tab. 5.1 (obtained with the default RTULF $= 10^{-5}$) shows that reducing the bound for turbulence variables may ensure values for lift and drag independent of the number of multigrid levels.

Influence of Limiter on Restart from Singlegrid
The effect of improving the converge properties by reducing the limiting factors for k and ω has also been checked for restarting from singlegrid. The singlegrid solution at Re $= 6.5 \cdot 10^6$ has been obtained after $40\,000$ iterations using EPSC $= 0.2$ together with RTULF $= 10^{-10}$. Restarting from this solution with different numbers of multigrid levels does not show the behaviour discussed before (Sect. 5.1.3.3), but exhibits an undisturbed behaviour comparable to the results based on the application of the Baldwin/Lomax turbulence model. Close–ups of the first iterations after restarting are presented in Figs. 5.40 – 5.42. Thus the multigrid method in the FLOWer code is able to treat turbulence equations correctly, too.
One reason for this limiter dependent behaviour when restarting from singlegrid was found to be the forcing function. The k residual seems to be the most crucial one since its convergence behaviour is more limiter dependent than the convergence behaviour of the ω residual. The density residual and the ω residual are disturbed due to the coupling of the k equation to the RANS equations as well as to the ω equation.

The forcing function as given in Eq. (3.96) is defined via the residual, i. e. the complete

spatial discretization including the source term, computed from the conservative variables after one Runge–Kutta cycle on the finest grid. From this approach the fine grid residuals of the RANS equations used for the forcing function are approximately the same as the change of the mean variables over the last time step:

$$R_{i,j}^{n+1} \approx \frac{W_{i,j}^{n+1} - W_{i,j}^{n}}{\tau_{i,j}} \,. \tag{5.3}$$

For the turbulence equations these residuals might yield negative values when the turbulence quantities are updated. Since the turbulence variables — and not the residuals — are bounded from below the fine grid residuals used for the forcing function generally differ from the change of the turbulence variables over the last time step:

$$R_{tu;\,i,j}^{n+1} \neq \frac{W_{\mathrm{lim};\,i,j}^{n+1} - W_{\mathrm{lim};\,i,j}^{n}}{\tau_{i,j}} \,. \tag{5.4}$$

Thus the forcing function for the turbulence equations might be unequal to (machine) zero leading to nonzero coarse grid corrections although the value of the turbulence variables did not change over the last time step on the finest grid.
Modifications of the forcing function as the crucial point of the nonlinear version of multigrid method, the full approximation scheme, were not found to yield controllable results.

5.1.3.3.4 Optimized Settings

Searching for optimized combinations of `RTULF`, the number of multigrid levels and the type of multigrid cycle at different Reynolds numbers was unsuccessful: At $\mathrm{Re} = 100 \cdot 10^6$ for no number of multigrid levels larger than one a combination of the type of multigrid cycle and the value of `RTULF` was found leading to some convergence of any residual.

5.1.3.4 Summary

In the previous sections the convergence behaviour using the k–ω turbulence model of Wilcox [131, 132] has been investigated on the same test cases — viscous flow around RAE 2822 airfoil at flow conditions of experimental test case no. 9 [25] using different Reynolds numbers on correspondingly adapted grids — as used for the investigation of the convergence behaviour applying the algebraic turbulence model of Baldwin and Lomax in Sect. 5.1.2.

It has been found that the application of multigrid to the turbulence equations leads to degraded convergence behaviour when increasing the amount of coarse grid operations by taking into account more multigrid levels as well as by using a W–cycle instead of a V–cycle. It has been shown that this effect is amplified at larger Reynolds numbers.
The perturbation of convergence observed when applying multigrid to the k–ω equations has partly its reason in the disjunction of the time rate of change of the turbulence quantities on the one hand, and the residual used for the forcing function on coarse grid levels calculated as the spatial discretization of the turbulence equations on the

finest grid on the other.
Although improvement of the level of convergence has been obtained in several cases by reducing the limiting factors for the turbulence quantities k and ω these factors are not allowed to meet case dependent values in order to ensure stability.

By parameter variations it has been verified that the multigrid method as implemented in the FLOWer code works correctly for turbulence equations. However, only by reduction — or even neglection — of multigrid it was possible to converge of all residuals down to machine accuracy for all test cases under investigation.

5.1.4 k–ω Calculations of Turbulence Decay

In this section the influence of one part of the turbulence source terms, the destruction term, is investigated. For this another test case, the simulation of free turbulent flow i. e. the decay of turbulence in uniform mean flow as considered e. g. by Rudnik [105] is taken.
The test case is described in detail in Sect. 5.1.4.1 whereas the influence of the number of multigrid levels on the convergence of the density residual together with the k and the ω residual is discussed in Sect. 5.1.4.2. This section is closed by a brief summary in Sect. 5.1.4.3.
Different to the investigations of the RAE 2822 airfoil the multigrid restart from a singlegrid solution is not investigated since the multigrid method has been shown to work correctly. Furthermore the influence of the limiting factor `RTULF` on the convergence behaviour is not discussed as the computational domain is too small to allow the turbulence quantities to decay close to their bounds.

5.1.4.1 Description of Free Turbulent Flow Test Case

This test is usually used to check the correctness of turbulence models as the decay of turbulence at uniform mean flow is computed. Therefore an equidistant grid with no other boundary conditions except for freestream boundaries is used. Furthermore the calculation is completely explicit i. e. the implicit residual smoothing is switched off. The same holds for the implicit smoothing of multigrid corrections i. e. `EPSC` $= 0.0$.
The flow conditions are $\mathbf{M}_\infty = 0.3$ with angle of attack $\alpha = 0^\circ$ and the Reynolds number $\mathrm{Re} = 10 \cdot 10^6$. Additionally the ratio of laminar to turbulent viscosity, $\mu_{l;\infty}/\mu_{tu;\infty}$ is given as `RLTU` $= 0.1639443$ at inflow. The limiting factor of the turbulence variables is set to `RTULF` $= 10^{-8}$.
The mean flow is laminar throughout the flow field as no gradients exist orthogonal to the flow direction. Since the decay of turbulence from inflow to outflow is investigated minor gradients of the mean flow variables in flow direction occur. The contribution of the turbulent kinetic energy k to the total energy E as well as changes in the eddy viscosity cause these gradients. They determine the production of turbulence (Eq. (2.42) resp. Eq. (2.44)) via the strain rate invariant $\mathbf{S}$ as given in Eq. (2.43) and via the divergence of velocity. As the only gradient to be considered is $\partial u/\partial x$ the production term of k reduces to

$$P_k = \mu_{tu} \frac{4}{3} \left(\frac{\partial u}{\partial x} \right)^2 - \frac{2}{3} \rho k \frac{\partial u}{\partial x} \,. \tag{5.5}$$

Since $\partial u/\partial x \ll 1$ can be taken as granted throughout the flow field the production term becomes negligibly small. Since ω is subject to a lower bound $\omega \gg 2\partial u/\partial x$ holds, too. Thus the production is close to zero but negative for this test case. That is both the k and the ω source term are only destructive within this test case. The physical phenomenon represented is the equilibrium of convection and destruction of turbulence.

5.1.4.2 k–ω Calculations Using Different Numbers of Multigrid Levels

At first it has to be stated that in case of singlegrid calculation the density residual equals zero for all iterations. Thus it is not visible in Fig. 5.43 where the density residual for two to four multigrid levels in a W–cycle is plotted. This residual equals zero in the first iteration for any number of multigrid levels since the first iteration corresponds to the first Runge–Kutta step on the finest grid i. e. a singlegrid iteration. From the second iteration on the density residual converges from below 10^{-10} down to machine accuracy. As multigrid theory proclaims coarse grid corrections equal to zero if and only if the fine grid residual equals zero, these non–zero coarse grid corrections have not been clear in the beginning. In Sect. 6.2.1 a different coupling strategy between mean and turbulence equations is introduced. This coupling was found to yield the non–zero density residuals when using more than one multigrid level.
The k as well as the ω residual converge to zero applying only singlegrid. Using more than one multigrid level leads to convergence over 17 orders of magnitude as shown in Fig. 5.44 and Fig. 5.45 respectively.

For the density residual the influence of multigrid disturbs the convergence behaviour as already noted above. For the k residual as well as for the ω residual a convergence improving effect of multigrid can be seen only for a two level cycle. Using three or four multigrid levels (nearly) no further improvement has been obtained.

5.1.4.3 Summary

This test case has been chosen for the investigation of a destructive source term's influence on the performance of a multigrid method. No destabilizing effects have been found although no further speed–up has been obtained if more than two multigrid levels have been used.
Thus it seems that not a source term in general but productive source terms give rise to robustness problems as encountered in Sect. 5.1.3. Destructive source terms seem to degrade multigrid efficiency.

5.1.5 Conclusions from Numerical Observations

It has been shown that generating grids with special emphasis on the adaptation of relevant boundary layer related lengths — δ resp. y^+ — yields well resolved velocity gradients within the boundary layer at Reynolds numbers ranging from wind tunnel to flight Reynolds numbers.
Based on these grids numerical investigations of the convergence behaviour of the

Reynolds averaged Navier–Stokes equations have been performed. Using an algebraic turbulence model (by Baldwin and Lomax [4]) firstly no convergence problems occurred: Neither the Reynolds number together with the corresponding Re–adapted grid nor the multigrid cycle applied yielded any robustness problems. Only a slowdown in convergence due to the increase of the cell aspect ratio over the Reynolds numbers has been observed.
Thus it is concluded that the RANS equations themselves can be integrated robustly over a wide range of Reynolds numbers using an explicit Runge–Kutta method accelerated by multigrid, local time stepping and implicit residual smoothing.

Switching to a k–ω turbulence model used for the RANS calculations an analogous slowdown over the Reynolds numbers has been observed, but in singlegrid convergence. Speeding up the convergence by multigrid lead to more and more serious perturbations of convergence for increasing Reynolds numbers. These perturbations have been found to be intensified by increasing the work done on coarser multigrid levels, i. e. using more multigrid levels as well as applying a W–cycle instead of a V–cycle. At $\mathrm{Re} = 100 \cdot 10^6$ no convergence at all could be achieved except for singlegrid. Some influence of the limiting factors $\lim_k$ and $\lim_\omega$ — represented in the FLOWer code by `RTULF` — on the convergence behaviour has been detected. Although in several cases improvements could be obtained these parameters were not found to be crucial.
From these results it is concluded that turbulence source terms may seriously disturb the convergence in case of multigrid.

No convergence problems occurred when calculating turbulence decay with different numbers of multigrid levels. The observed effect in the residuals if switching multigrid on, i. e. convergence to zero for singlegrid but "only" machine accuracy for multigrid computations, probably comes from the coupling between RANS and turbulence equations. This conclusion is corroborated by results obtained for varying `RTULF` since the convergence behaviour of the density residual has been influenced via the value of `RTULF`.
As turbulence decay represents the equilibrium between convection and destruction of turbulence it is concluded that source terms do not cause problems in combination with multigrid in general. Destructive source terms were found to reduce multigrid efficiency but not to disturb convergence.

Summarizing the above results productive source terms of the k–ω turbulence model equations are found to oppose standard multigrid techniques.

5.2 Fourier Analysis of Model Problem Equation including Source Term

The results from Fourier analysis for a scalar convection–diffusion equation in Sect. 4.5 have been corroborated by the RANS simulations using an algebraic turbulence model presented in Sect. 5.1.2.
In this section the effects arising from turbulence source terms as shown in Sect. 5.1.3 resp. Sect. 5.1.4 will be analyzed by means of the Fourier analysis introduced in Chap. 4 now including the source term.

Therefore eigenvalues of the spatial discretization operators are derived in Sect. 5.2.1. Sect. 5.2.2 and Sect. 5.2.3 present some Fourier analysis results for different values of the source term coefficient S in relation to the eigenvalues derived before. Conclusions are drawn in Sect. 5.2.4.

5.2.1 Eigenvalues of Spatial Discretization Operators

In order to give a statement on the numerical behaviour of the discretized model problem equation it is necessary to know the eigenvalues of the spatial discretization operators $Z_c\circ$, $Z_v\circ$ and $Z_S\circ$ since renumbering the grid points in a consecutive way e. g.

$$(i,j) \rightarrow n = i + i_{\max}\cdot(j-1) \quad ; \quad n_{\max} = i_{\max}\cdot j_{\max}$$

and applying $Z\circ = Z_c\circ + Z_v\circ + Z_S\circ$ to each grid point yields a $n_{\max}\times n_{\max}$ matrix $\mathbf{Z}$ which represents the discrete differential operator $Z\circ$ in matrix notation. A particular solution of

$$\begin{aligned}
\left(\frac{\partial W}{\partial t}\right)_{n=1,\ldots,n_{\max}} &= \\
\left(\frac{\partial W}{\partial t}\right)_{\substack{i=1,\ldots,i_{\max}\\ j=1,\ldots,j_{\max}}} &= (Z\circ W)_{\substack{i=1,\ldots,i_{\max}\\ j=1,\ldots,j_{\max}}} \\
&= (Z\circ W)_{n=1,\ldots,n_{\max}} \\
&= \mathbf{Z}\cdot(W)_{n=1,\ldots,n_{\max}}
\end{aligned} \tag{5.6}$$

is

$$e^{\mathbf{Z}\cdot t}\,.$$

Thus the eigenvalues of $\mathbf{Z}$ resp. of $Z\circ$ influence the behaviour of the time dependent solution. Since time is used as some kind of iteration counter to approach steady state the numerical method always deals with "time dependent" solutions independent of their physical meaning.

As has been done in Chap. 4 constant values for u, v and μ as well as for Δx, Δy and the source term coefficient S over the whole flow field are assumed. By this assumption the matrix $\mathbf{Z}$ becomes a Toeplitz matrix, i. e. on each line parallel to the main diagonal all elements are equal. Vichnevetsky and Bowles [128] note that

$$\left\{e^{\mathbf{I}n\Phi}\right\}_{n=1,\ldots,n_{\max}}$$

with

$$\begin{aligned}
n\Phi &= (i,j)\bullet\begin{pmatrix}\Phi_x\\ \Phi_y\end{pmatrix} \\
&= i\Phi_x + j\Phi_y
\end{aligned}$$

are the eigenvalues of any Toeplitz matrix. From this they derive that the eigenvalues of discrete differential operators are their Fourier symbols. Here the eigenvalues of $Z\circ$ are $\mathcal{Z}(\Phi_x,\Phi_y)$.

Due to $\mathcal{Z}_c$, Eq. (4.17), the Fourier symbol $\mathcal{Z}$ is complex valued. While the imaginary part is related to the phase velocity error, i. e. dispersion, introduced by the spatial discretization the real part of a Fourier symbol acts on the amplitude of the Fourier modes $e^{\mathbf{I}i\Phi_x+\mathbf{I}j\Phi_y}$. As dispersive effects are not of interest within this work the imaginary part is not taken into account for the further investigation.
Considering the real part of the Fourier symbol $\mathcal{Z}(\Phi_x, \Phi_y)$ three different cases have to be investigated:

- $\mathbf{Re}\left[\boldsymbol{\mathcal{Z}}\ \boldsymbol{(\Phi_x, \Phi_y)}\right] < \mathbf{0}$ for all $(\Phi_x, \Phi_y) \in [-\pi, \pi]^2$,
- $\mathbf{Re}\left[\boldsymbol{\mathcal{Z}}\ \boldsymbol{(\Phi_x, \Phi_y)}\right] = \mathbf{0}$ for one pair of phase angles (Φ_x, Φ_y) and
- $\mathbf{Re}\left[\boldsymbol{\mathcal{Z}}\ \boldsymbol{(\Phi_x, \Phi_y)}\right] > \mathbf{0}$ for at least some phase angle pairs (Φ_x, Φ_y).

The latter two cases coincide in this work as the matrix $\mathbf{Z}$ does not become singular necessarily if $\mathrm{Re}\left[\mathcal{Z}\ (\Phi_x, \Phi_y)\right] = 0$ as the imaginary part of $\mathcal{Z}(\Phi_x, \Phi_y)$ equals zero only if

$$u\Delta y \sin(\Phi_x) = -v\Delta x \sin(\Phi_y)\,. \tag{5.7}$$

Using the values of u, v, Δx and Δy given in Eq. (4.73) this is equivalent to

$$\sin(\Phi_x) = -1000 \sin(\Phi_y)\,. \tag{5.8}$$

This equation can be fulfilled for $|\sin(\Phi_y)| \leq 0.001$ only. This condition leads to $\Phi_y \leq 0.001$. Thus Φ_y has to be much smaller than the resolution of phase angles in use: $\Delta\Phi = \pi/90 \approx 0.035$ i. e. these Fourier modes are not represented within this analysis. More than 3000 points — instead of 90 points — per direction would be necessary to resolve these Fourier modes.

Analogously the model problem can be treated as an eigenvalue problem with the source term coefficient S as the parameter of interest. This approach has been used e. g. by Trottenberg et. al. [118], Elman et. al. [35] or Brandt and Livshits [13] discussing the multigrid treatment of the Helmholtz equation, a diffusion equation with a source term.

5.2.2 Influence of Destructive Source Term on Amplification Factor

In this section the influence of destructive source terms on the damping properties of the numerical method introduced in Chap. 3 is investigated by means of a Fourier analysis.

A destructive source term is given by $S < 0$. This implies $\mathcal{Z}_S < 0$ which leads to

$$\begin{aligned} \mathrm{Re}\left[\mathcal{Z}\ (\Phi_x, \Phi_y)\right] &= -u\Delta y\,[1 - \cos(\Phi_x)] - v\Delta x\,[1 - \cos(\Phi_y)] \\ &\quad - 2\mu\frac{\Delta x}{\Delta y}\,[1 - \cos(\Phi_y)] + \mathcal{V}S \\ &< 0\,. \end{aligned} \tag{5.9}$$

This is valid for any pair of phase angles $(\Phi_x, \Phi_y) \in [-\pi, \pi]^2$.
Two different investigations have been performed: In Sect. 5.2.2.1 the influence of an increasingly destructive source term on the amplification factor $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$ of a Runge–Kutta method without using multigrid as given in Eq. (4.50) is discussed. Application of multigrid to the model problem subject to a destructive source term is treated in Sect. 5.2.2.2. In Sect. 5.2.2.3 some conclusions are given.

5.2.2.1 Damping Properties of Runge–Kutta Method acting on Model Problem Equation including Destructive Source Term

The damping properties of the Runge–Kutta method for a destructive source term are investigated in two ways: On the one hand the source term coefficient has been changed from $S = -10^0$ up to $S = -10^5$. The corresponding amplification factor averages are listed in Tab. 5.4. On the other hand the Fourier footprint for $S = -10^3$ is discussed in more detail as an example.

For $S < -1000$ the Fourier footprints would show only minor differences compared to the reference in Fig. 4.6. Although nearly invisible in the Fourier footprints the overall damping properties of the Runge–Kutta method are improved due to the increasingly destructive source term as summarized in Tab. 5.4 which is discussed at the end of this section.

From Fig. 5.46, the Fourier footprint of $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$ for $S = -1000$, differences to the reference Fourier footprint, Fig. 4.6, can be easily derived: Frequencies close to $(\Phi_x, \Phi_y) = (0, 0)$ experience a significantly improved damping by an amplification factor less than 0.9. Another frequency area with improved damping is around $(\Phi_x, \Phi_y) = (0, \pm\pi)$. On the other hand the high–frequency modes in the vicinity of $(\Phi_x, \Phi_y) = (\pm\pi, \pm\pi)$ are less efficiently damped: The amplification factor is raised to nearly 0.3 for high–frequency modes being damped by about 0.2 if no source term is present. Or the amplification factor climbs from less than 0.3 to 0.35 for some other high–frequency modes. This effect can be counteracted by changing the parameters of the implicit residual smoothing in x–direction as given in Sect. 4.5. Reducing the influence of the implicit smoothing in x–direction by lowering the bound for the coefficients of implicit smoothing to 0.1 instead of 0.2 yields the Fourier footprint plotted in Fig. 5.47. Although the average of $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$ over all modes is improved by nearly 20% changing the coefficients interval to $[0.1, 1.2]$ the choice of smoothing coefficients for turbulence equations is not treated here in more detail as this is an additional topic.
The effect of implicit smoothing on high–frequency modes indicates why the overall averaged amplification factor $\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right|_{\mathrm{all}}$ is slightly increased compared to the reference case without source term as shown in Tab. 5.4.

For $S = -10^4$ resp. $S = -10^5$ the model equation (Eq. (4.1)) is no longer dominated by its convection or diffusion part. $S = -10^4$ yields — in combination with the specified settings for u, v, μ, Δx and Δy — that the Fourier symbol of the source term $\mathcal{Z}_S$ is of the same order of magnitude

$$\mathcal{Z}_S = -10^{-4} \tag{5.10}$$

as the sum of the Fourier symbols of convection and diffusion at its maximum absolute value,

$$\mathcal{Z}_c + \mathcal{Z}_v \approx -3 \cdot 10^{-4}\,. \tag{5.11}$$

From the values of $\overline{\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right|}_{\mathrm{all}}$ in Tab. 5.4 the conclusion arises that the Runge–Kutta method in use is not suited for source term dominated flows. This conclusion can be substantiated by corresponding Fourier footprints but these are not presented here.
Since ω is proportional to $1/y_n^2$ — with y_n denoting the distance in wall normal direction — close to walls [105, 131] and the destructive part of the source term of the ω–equation, Eq. (2.45), depends on ω^2 comparably large destructive source term contributions are not unrealistic in applications. Still it was not possible to prove convergence problems caused by source term dominated flow regions.

source term coefficient	$\overline{\left\|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\|}_{\mathrm{all}}$	$\overline{\left\|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\|}_{\mathrm{low}}$	$\left\|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\,(0,0)\right\|$
0	0.37094	0.51023	1.00000
-1	0.37094	0.51019	0.99985
-10	0.3709	0.50986	0.9985
-100	0.37058	0.50668	0.98522
-1000	0.37613	0.48423	0.86959
-10000	0.52609	0.45797	0.40005
-100000	0.66434	0.45993	0.06251

Table 5.4: Amplification factor averages and its absolute value for $(\Phi_x, \Phi_y) = (0,0)$ related to the source term coefficient S (destruction) — singlegrid analysis

In Tab. 5.4 the amplification factor averages of the Runge–Kutta method, Eq. (4.50), over all Fourier modes and over all low–frequency Fourier modes as defined in Eq. (4.68) resp. Eq. (4.69) are given for different destruction representing values of the source term coefficient S. Additionally the absolute value of $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$ for the Fourier mode $(\Phi_x, \Phi_y) = (0,0)$ is noted in the table.

Increasing destruction leads to a more and more improved damping of the Fourier mode $(\Phi_x, \Phi_y) = (0,0)$ which is undamped in a pure convective–diffusive flow ($S = 0$). For this mode the amplification factor equals

$$g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\,(0,0) = 1 + \tau \frac{1}{1 - \tau \min\left(0, \mathcal{Z}_S\right)} \mathcal{Z}_S = \frac{1}{1 - \tau \mathcal{Z}_S} \tag{5.12}$$

as $\mathcal{B}(0,0) = 1$ and $\mathcal{Z}_v(0,0) = 0 = \mathcal{Z}_c(0,0)$. The latter equality yields — since $\alpha_5 = 1$ —

$$f^{(q=5)}_{\mathrm{pi-src,\,IRS}}\,(0,0) = \frac{\alpha_5}{1 - \alpha_5 \tau \min\left(0, \mathcal{Z}_S\right)} = \frac{1}{1 - \tau \min\left(0, \mathcal{Z}_S\right)}\,.$$

At the same time the amplification factor average over low frequencies is improved, too. An exception has to be made for source term dominated flow conditions ($S = -10^5$).
The average of the amplification factor over all frequencies exhibits an improvement only for small absolute values of the source term coefficient S. In case of increasing source term dominance the averaged amplification factor is nearly doubled compared to convection diffusion dominated flow situations.

5.2.2.2 Damping Properties of Multigrid Method acting on Model Problem Equation including Destructive Source Term

Based on the results presented above $S = -1000$ has been taken as a showcase destructive source term for the investigation of its influence within a multigrid Fourier analysis.

The Fourier footprint for a two level cycle in Fig. 5.48 has an even more significantly different appearance compared to its reference ($S = 0$) in Fig. 4.8 than the singlegrid footprint for $S = -1000$ (Fig. 5.46) in relation to Fig. 4.6.
In Fig. 5.48 the damping properties have been improved around $(\Phi_x, \Phi_y) = (0, \pm\pi)$ further. However, most of this improvement is already visible in Fig. 5.46, the corresponding plot using only one multigrid level. Another region of reduced amplification factor values is in the vicinity of $\Phi_y = 0$, especially close to $\Phi_x = 0$. This is due to the

# multigrid level	$\overline{\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\rvert}_{\mathrm{all}}$	$\overline{\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\rvert}_{\mathrm{low}}$	$\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\,(0,0)\right\rvert$
1	0.37613	0.48423	0.86959
2	0.3136	0.32122	0.57979
3	0.30452	0.29009	0.23199
4	0.30332	0.28561	0.04642

Table 5.5: Amplification factor averages and its absolute value for $(\Phi_x, \Phi_y) = (0, 0)$ related to the number of multigrid levels considering a destructive source term coefficient ($S = -1000$) — multigrid analysis

amplification factor in $(\Phi_x, \Phi_y) = (0, 0)$ which is about 0.58. As listed in Tab. 5.5 this value gets smaller and smaller with an increasing number of multigrid levels. In the vicinity of $(\Phi_x, \Phi_y) = (\pm\pi, \pm\pi)$ the influence of multigrid is not relevant as expected. Here the results obtained in the section above applying only one multigrid level still hold.

In Figs. 5.49 – 5.50 — showing Fourier footprints considering $S = -1000$ using three or four multigrid levels — only minor differences except for the neighborhood of the Fourier mode for $(\Phi_x, \Phi_y) = (0, 0)$ can be detected compared to Fig. 5.48.

Comparing Tab. 5.5 with Tab. 4.1 results in the observation that on one side the average amplification factor over all Fourier modes is less decreased with increasing number of multigrid levels in the presence of a destructive source term. This observation is valid for the absolute as well as relative decrease. The latter is less than 20% from one to four multigrid levels considering $S = -1000$ (cmp. Tab. 5.5) while the relative decrease over the number of multigrid levels for $S = 0$ (cmp. Tab. 4.1) is close to 23%.
The reason for this reduced efficiency lies in the damping of high–frequency Fourier modes. As an example Fig. 4.10 and Fig. 5.50 are compared. The latter figure shows the damping properties of a four level V–cycle acting on a destructive source term while the first gives the corresponding results without a source term. The run of the isoline 0.25 in Fig. 4.10 encloses about 50% of all Fourier modes. That is they are damped with $\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\,(\Phi_x, \Phi_y)\right\rvert \leq 0.25$. The corresponding isoline in Fig. 5.50 looks the same for small absolute values of Φ_x and Φ_y. But for Φ_x and Φ_y representing high–frequency

modes the runs differ considerably as already discussed in Sect. 5.2.2.1 using singlegrid only.
The corresponding relative decrease of the amplification factor averaged over the low frequencies, $\overline{\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right|}_{\mathrm{low}}$, is larger than 40% for both source term coefficients discussed here.
For the source term coefficient $S = -1000$ both amplification factor averages exhibit a reduced influence of the fourth multigrid level compared to the averages without a source term. This can be traced back to the values of the Fourier symbols of spatial discretization operators on the coarsest multigrid level: Calculating the values for Δx and Δy on the fourth multigrid level based on Eq. (4.73) yields

$$\mathcal{V} = 64 \cdot 10^{-8} \tag{5.13}$$

for the volume of a cell on this level. The Fourier symbol of the spatial discretization of the source term, Eq. (4.29), equals

$$\mathcal{Z}_S = \mathcal{V} \cdot S = -6.4 \cdot 10^{-4} \tag{5.14}$$

for $S = -1000$ on the fourth grid level. The sum of the Fourier symbols for the convection and the diffusion terms on the same level reaches the maximum of its absolute value at about the same magnitude:

$$\mathcal{Z}_c + \mathcal{Z}_v \approx -10 \cdot 10^{-4}\,. \tag{5.15}$$

In analogy to the results obtained for singlegrid analysis the flow tends to become source term dominated on the fourth multigrid level leading to a decrease in the overall damping properties.
This result may be an explanation for the degraded multigrid efficiency for the turbulence decay simulation reported in Sect. 5.1.4.2.

5.2.2.3 Summary

In the previous sections it has been shown by means of Fourier analysis that destructive source terms improve the damping properties of a Runge–Kutta method especially for low–frequency Fourier modes in comparison to a convection–diffusion equation without a source term. This holds as long as the source term does not dominate over convection and diffusion. Mainly the amplification factor for high–frequency modes is increased for source term dominated flows.
The improving effect of a destructive source term on low–frequency modes is intensified by the application of multigrid. As the Fourier symbol of the source term scales with the volume of a control cell it grows faster than the symbols of convection and diffusion going to ever coarser grid levels. This has been found to yield source term dominated flows on the coarsest multigrid levels. This effect reduces the efficiency of multigrid in the presence of destructive source terms.

5.2.3 Influence of Productive Source Term on Amplification Factor

Analogously to Sect. 5.2.2 in this section the influence of a source term on the damping properties of the baseline numerical method is investigated by means of a Fourier

analysis. While destructive source terms have been considered the productive source terms are looked at here.

A productive source term is given by $S > 0$. This implies $\mathcal{Z}_S > 0$ which leads to

$$\begin{aligned} \mathrm{Re}\left[\mathcal{Z}\left(\Phi_x, \Phi_y\right)\right] &= -u\Delta y\left[1-\cos(\Phi_x)\right] - v\Delta x\left[1-\cos(\Phi_y)\right] \\ &\quad - 2\mu\frac{\Delta x}{\Delta y}\left[1-\cos(\Phi_y)\right] + \mathcal{V}S \\ &> 0 \quad \text{for some } (\Phi_x, \Phi_y) \in [-\pi, \pi]^2 \ . \end{aligned} \tag{5.16}$$

Again two different investigations have been performed: In Sect. 5.2.3.1 the influence of an increasingly productive source term on the amplification factor $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$ of a Runge–Kutta method without using multigrid as given in Eq. (4.50) is discussed. Application of multigrid to the model problem subject to a productive source term is treated in Sect. 5.2.3.2. In Sect. 5.2.3.3 some conclusions are given.

5.2.3.1 Damping Properties of Runge–Kutta Method acting on Model Problem Equation including Productive Source Term

In analogy to Sect. 5.2.2.1 the damping properties of the Runge–Kutta method for a productive source term are investigated in two ways: Amplification factor averages for a range of productive source term coefficients — from $S = 10^0$ up to $S = 10^5$ — are listed in Tab. 5.6. On the other hand the exemplary Fourier footprint for $S = 10^3$ is investigated more closely.

For any positive source term coefficient Fourier footprints would differ from the reference plot (Fig. 4.6) in the vicinity of $(\Phi_x, \Phi_y) = (0, 0)$. There is a region where the amplification factor is larger than 1.0. Although this region is considerably small for $S = 1$ both averaged amplification factors are slightly increased as given in Tab. 5.6. This region of amplified Fourier modes is enlarged along $\Phi_y = 0$ for an increased productive source term. Except for this no further differences to Fig. 4.6 occur.

First changes outside the vicinity of $\Phi_y = 0$ can be observed for $S = 1000$ in Fig. 5.51. Here the amplification factor differs from the reference in Fig. 4.6 also for modes close to $(\Phi_x, \Phi_y) = (0, \pm\pi)$. In contrast to the overall decline of the damping properties they are improved in these regions. In the vicinity of $\Phi_y = 0$ the amplification factors of modes with $|\Phi_x| \lessapprox \pi/2$ are raised above 1.1 for $S = 1000$.
The averages of the amplification factor over all modes as well as over the lower frequencies only are increased by about 8% resp. 12.5% compared to the reference without a source term as noted in Tab. 5.6. For less productive source terms the averages have been increased by not more than 1.5%.

As for the destructive source term $S = -1000$ the influence of the smoothing coefficient range has been checked for $S = 1000$. From Fig. 5.52 it can be seen that the damping properties surrounding $\Phi_y = 0$ are getting worse. While for $\varepsilon_i, \varepsilon_j \in [0.2, 1.2]$ the amplification factor exceeds 1.1 only for $|\Phi_x| \lessapprox \pi/2$ (cmp. Fig. 5.51) now — for $\varepsilon_i, \varepsilon_j \in [0.1, 1.2]$ — the amplification factor gets larger than 1.1 for all $\Phi_x \in [-\pi, \pi]$.

source term coefficient	$\overline{\left\vert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\vert}_{\mathrm{all}}$	$\overline{\left\vert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\vert}_{\mathrm{low}}$	$\left\vert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\,(0,0)\right\vert$
0	0.37094	0.51023	1.0000
1	0.37097	0.51029	1.0001
10	0.37125	0.51086	1.0015
100	0.37399	0.51657	1.015
1000	0.40208	0.57503	1.15
10000	0.72906	1.2202	2.4997
100000	4.4676	7.9998	15.997

Table 5.6: Amplification factor averages and its absolute value for $(\Phi_x, \Phi_y) = (0,0)$ related to the source term coefficient S (production) — singlegrid analysis

For destruction source term dominance occurred for source term coefficients less than -10000. Dominance of a productive source term is gained for $S = 100000$ at first. Then all modes are amplified over one time step what is indicated by the value of $\overline{\left\vert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\vert}_{\mathrm{all}}$ at this source term coefficient in Tab. 5.6.

Tab. 5.6 is the counterpart to Tab. 5.4 but for different productive source terms. For the phase angle $(\Phi_x, \Phi_y) = (0,0)$ the amplification factor now equals

$$g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\,(0,0) = 1 + \tau \frac{1}{1 - \tau \min(0,\ \mathcal{Z}_S)} \mathcal{Z}_S = 1 + \tau \mathcal{Z}_S\,. \tag{5.17}$$

The values of $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$ at $(\Phi_x, \Phi_y) = (0,0)$ contain a conspicuous "15". This "15" moves more and more to the left with increasing production. It is introduced by the value of the time step per volume τ being close to $7.5 \cdot 2000 = 15000$.
The right hand side of Eq. (5.17) is the reciprocal of the right hand side of Eq. (5.12) as here $\mathcal{Z}_S$ is a positive number while Eq. (5.12) is valid for negative values of $\mathcal{Z}_S$ only. Thus the values of $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\,(0,0)$ hold the equation

$$\left\vert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\,(0,0;\ S)\right\vert = \frac{1}{\left\vert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\,(0,0;\ -S)\right\vert}\,.$$

In the same way Tab. 5.4 shows more and more improved damping on $(\Phi_x, \Phi_y) = (0,0)$ for increasingly destructive source terms, Tab. 5.6 presents an increasing amplification for this mode with growing production. A corresponding boosting effect is found for the averages over all and over low frequencies respectively of the amplification factor $\overline{\left\vert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\vert}$ different from the non–monotone changes of these averages over increasing destruction.

5.2.3.2 Damping Properties of Multigrid Method acting on Model Problem Equation including Productive Source Term

In analogy to Sect. 5.2.2.2 $S = 1000$ is chosen as source term coefficient to investigate the effect of multigrid on a convection diffusion equation with a productive source term.

Fig. 5.51 shows the Fourier footprint of a singlegrid iteration and has already been discussed above. The Fourier footprint representing the damping properties of a two level cycle at $S = 1000$ is given in Fig. 5.53. For $|\Phi_y| \gtrapprox \pi/4$ two effects combine: the improvement of damping properties close to $\Phi_x = 0$ due to the application of multigrid — compare Fig. 4.8 to Fig. 4.6 — is one of them. The other effect is the influence of the productive source term on these modes as shown in Fig. 5.51. That is for $(\Phi_x, \Phi_y) \approx (0, \pm\pi)$ the damping properties are improved further and the modes $(\Phi_x, \Phi_y) = (\pm\pi, \pm\pi)$ experience a decline in damping.

Around $\Phi_y = 0$ the Fourier modes become more amplified due to multigrid. As noted in Tab. 5.7 $\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\ (0,0)\right|$ is increased by 50% using a second multigrid level. Nevertheless the averages of $\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right|$ are both reduced by applying two levels.
At first glance the Fourier footprints using a three / four level cycle in Fig. 5.54 or

# multigrid level	$\overline{\left\|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\|}_{\mathrm{all}}$	$\overline{\left\|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\|}_{\mathrm{low}}$	$\left\|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\ (0,0)\right\|$
1	0.40208	0.57503	1.15
2	0.35137	0.47226	1.7248
3	0.35642	0.49477	4.3106
4	0.37813	0.5803	21.5416

Table 5.7: Amplification factor averages and its absolute value for $(\Phi_x, \Phi_y) = (0, 0)$ related to the number of multigrid levels considering a productive source term coefficient ($S = 1000$) — multigrid analysis

Fig. 5.55 do not offer new insights. This is partly due to the range of isolines chosen: no isolines on values larger than 1.2 have been plotted in order to keep a common color representation for all these plots.

In Tab. 5.7 the amplification factor averages $\overline{\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right|}_{\mathrm{all}}$ and $\overline{\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right|}_{\mathrm{low}}$ are listed together with $\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\ (0,0)\right|$ for different numbers of multigrid levels at $S = 1000$.
With an increasing number of multigrid levels $\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\ (0,0)\right|$ is boosted from 1.15 (singlegrid) up to about 21.5 applying a four level V–cycle. While the amplification factor averages are improved when switching from singlegrid to two level considering the third level yields a changed tendency: The averages $\overline{\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right|}_{\mathrm{all}}$ and $\overline{\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right|}_{\mathrm{low}}$ become larger applying an additional multigrid level. With four multigrid levels the average of the amplification factor over low frequencies exceeds its value for singlegrid.

In contrast to the idea of multigrid — damp low–frequency modes more efficiently on coarser grids where they have a higher frequency in relation to the grid resolution — a productive source term yields the opposite effect within multigrid: Increasing the number of multigrid levels at least very low–frequency modes are boosted more and more.

5.2.3.3 Summary

In the previous sections results from a Fourier analysis applied to the model problem including a productive source term have been presented. The damping properties of the Runge–Kutta method are significantly disturbed for low–frequency Fourier modes. These modes are amplified due to the productive source term.
As for destructive source terms this effect is intensified, too, — but much more significantly — if multigrid is applied. This contradicts the idea of multigrid: Instead of damping low–frequency modes on coarser grids more efficiently, these modes are amplified more and more.

5.2.4 Conclusions from Fourier Analysis

Fourier analysis results obtained for a 2D convection diffusion equation including a source term have been compared to those for the corresponding model problem without a source term. It has been shown that the influence of a source term differs depending on its sign.
Destructive source terms yielded an improved damping especially for low–frequency modes. This improvement has been found being intensified by the application of multigrid. However, it has been detected that the overall efficiency of multigrid is reduced in the presence of a destructive source term. This can be traced to a pseudo source term dominance on coarse multigrid levels.
On the other hand by productive source terms the damping properties for low–frequency modes are disturbed as these source terms lead to an amplification of these modes. Again the usage of multigrid intensifies this effect of the source term. Thus productive source terms in combination with multigrid result in an effect contradictory to the intended improvement of low–frequency modes' damping.

Based on these results — especially on destruction — I do not agree with Kunz and Lakshminarayana [69] on their statement that "the direct influence of [...] turbulence source terms can only be *de*stabilizing, for plausible values of [...] k, ε". In particular as they only use singlegrid for their investigations.
Further this analysis gives advise to reduce the influence of multigrid within the numerical treatment of transport equations including a source term.

5.3 Conclusions

The previous sections have presented results from investigations on the influence of turbulence source terms on the convergence behaviour of RANS simulations.

At first basic results from Fourier analysis as in Sect. 4.5 have been corroborated by Navier–Stokes calculations using a 0–equation turbulence model. Subsequently both numerical computations and Fourier analysis of source term containing transport equations have shown that the application of multigrid to these equations might yield destabilization of convergence.
Additionally it has been found that destructive source terms lead to a reduction of

multigrid efficiency although the singlegrid damping properties are improved. This reduced efficiency is based on a pseudo source term dominance on coarse multigrid levels. Productive source terms have been identified to destroy robustness even in combination with multigrid.

Since singlegrid computations using the k–ω turbulence model of Wilcox have not shown any stability problems at all, the conclusion to reduce or — if possible – to avoid multigrid for turbulence equations (as well as for any other [productive] source term containing equation) seems reasonable.
The following chapter discusses several approaches to do so. Reduction of multigrid might be gained by modifying prolongation operators. An implicit approach might be used in order to avoid multigrid for turbulence equations at all. Again numerical calculations as well as Fourier analyses are used to assess these approaches.

6 Improvements to the Numerical Treatment of Turbulence Equations

This chapter deals with modifications to the numerical method introduced in Chap. 3 in order to overcome the convergence problems discussed in Chap. 5.

During the preparation time of this thesis several approaches have been investigated. One of the first approaches has been to adjust the multigrid method to the turbulence equations. Secondly — since the results from modifying multigrid were not encouraging enough to pursue further — an implicit time integration of the turbulence equations has been implemented in the FLOWer code. Last but not least improvements to the coupling between mean and turbulence equations and to the time step for the turbulence equations have been made.
At first the improvements obtained for the multigrid treatment of the turbulence equations are described in Sect. 6.1. The adjustments made for the basic scheme are presented in Sect. 6.2. In Sect. 6.3 the implicit scheme chosen for implementation in the FLOWer code is introduced. It is presented last since it is a different time stepping scheme. Thus it enforced more complex changes to the FLOWer code. Conclusions are drawn in Sect. 6.4.

6.1 Improvements to Multigrid

As it has been found in Sect. 5.1.3, the application of multigrid to turbulence equations causes robustness problems. Adaptation of multigrid components to treat turbulence equations more effectively is expected to overcome these problems.
Some of the results shown in this section have already been presented in [37]. Here they are given with more emphasis on analyzing the effects of the modifications presented. Additionally circumstances are discussed in which the omission of multigrid updates is reasonable.

At first the components of the multigrid method have to be identified whose modification is expected to improve robustness of the whole method applied to the turbulence equations. As noted in Sect. 5.1.3.3.3 modifications of the forcing function did not yield controllable results. Still two kinds of operators are available for modifications: The fine resp. coarse grid operator on the one hand and the transfer operators on the other.

Both the fine grid and the coarse grid operator are built from a spatial discretization and a smoother. The spatial discretization of the turbulence equations is the same on

all grid levels: A first order upwind discretization is applied to the convection terms while a central discretization is used for the viscous fluxes as introduced in Sect. 3.1. The smoother within the multigrid method in the FLOWer code is a 5–stage Runge–Kutta time stepping scheme with local time stepping and implicit residual smoothing as presented in Sect. 3.2 and Sect. 3.3.1 resp. Sect. 3.3.2. This operator is already adapted to treat turbulence equations efficiently since source terms are taken into account point implicitly. Thus neither the fine grid operator nor the coarse grid operator are further adapted to the source term.

The restriction operator controls the transfer of data from a fine to a coarse grid. For a cell vertex discretization it is somehow fixed to be an injection for flow quantities as coarse grid points always coincide with fine grid points. The restriction of the residuals for the forcing function is performed by a volume weighted approach.
Sheffer et al. [109] introduce an underrelaxation factor to the restriction operator of the residuals. They claim to reduce instabilities on coarse grids due to collected and transferred chemical source terms by this underrelaxation.
One possibility found in literature to adapt the restriction operator to source terms is the approach suggested by Gerlinger et al. [40]. For chemically reacting flows they define a scaling factor for the restriction operator in order to reduce the transferred residuals and forcing functions in regions of high chemical activity:

$$\Theta = \max\left[0, 1 - \psi\theta\right] \tag{6.1}$$

θ is a source term based sensor

$$\theta = \left(\frac{1}{N_S - 1} \sum_{i=1}^{N_S - 1} \frac{|S_i|}{|S_i|_{\max} + \epsilon} \right)^{\alpha} \tag{6.2}$$

with the number of chemical species N_S and their source terms S_i. The maximum $|S_i|_{\max}$ of the absolute production of each species i is taken over the whole flowfield. ϵ is a small number to avoid division by zero. ψ in Eq. (6.1) is — as noted by Gerlinger et al. — a multigrid level dependent constant. Further they remark on the scaling factor and the sensor: "[...] A disadvantage of this method is the strong case dependence of the choice of parameters ψ and α to limit the degree of damping. [...]"
As this work is intended to obtain higher robustness for the application of the FLOWer code to high Reynolds number flows such a strong case dependence is to be avoided. Thus this approach has been deferred to a fall back case if other approaches fail. As will be shown below no necessity arose to fall back to this type of approach.
Both approaches to adapt the restriction operator modify the representation of the fine grid problem on the coarse grid by means of a global or local underrelaxation of the forcing function.

The remaining multigrid component to be optimized for turbulence equations is the prolongation operator which controls the transfer of the coarse grid corrections to fine grid points.
The standard method for refining coarse grid values in the FLOWer code is (bi–/tri–) linear interpolation as introduced in Sect. 3.3.3.4 and visualized in Fig. 3.7. As derived in the same section any prolongation operator for a cell vertex discretization can be split into a coarse–to–fine injection and a calculation of the values at fine grid points not coinciding with coarse grid points as schematically shown in Fig. 3.6.

In Sect. 6.1.1 a modification of the calculation of fine grid values is discussed. An improved prolongation operator based on a relaxation of the injection is investigated in Sect. 6.1.2. Sect. 6.1.3 has been added to discuss additional possibilities to modify the multigrid method. A summary of this section is given in Sect. 6.1.4.

6.1.1 Upwind Prolongation

The idea for this prolongation operator has been to develop a transfer operator that emulates in some way the upwind scheme used for integrating the convective part of the turbulence equations. By introducing an upwind character into the prolongation the transport of information against the local flow direction is avoided. This is physically correct as turbulence equations do not convect any acoustic waves.

Upwind directed prolongation operators for upwind discretized flows were already presented e. g. by Blazek [6] and Drikakis et al. [30]. Blazek used upwind prolongation for hypersonic flows only for $|\mathrm{M}| \geq 1$. Otherwise he applied a standard linear interpolation. Considering the differences between a cell centered and a cell vertex discretization the "upwind piece-wise constant prolongation" published by Drikakis et al. [30] is the same as the (unweighted) upwind directed prolongation introduced below.

6.1.1.1 Upwind Directed Prolongation

The upwind directed prolongation operator is derived in the following section. The analysis of its properties for a 1D example function is given subsequently. This section ends with some conclusions on this prolongation approach.

6.1.1.1.1 Derivation of Upwind Directed Prolongation

For this operator the injection of coarse grid values to fine grid points, Eq. (3.107), has not been altered. Only the calculation of values at intermediate fine grid points, Eqs. (3.111) – (3.112), is modified.
The first idea was to calculate the same upwind coefficients as is done for the upwind flux

$$\Delta W_{h,\,2i+1,2j} = \mathrm{M}_{2i+1,2j} \Delta W_{h,\,2i,2j} + (1 - \mathrm{M}_{2i+1,2j})\, \Delta W_{h,\,2i+2,2j} \tag{6.3}$$

but this just yields a linear interpolation with variable coefficients instead of symmetric weights. This can be seen by comparing Eq. (6.3) to

(3.110) $$\Delta W_{h,\,2i+1,2j} = a^- \Delta W_{h,\,2i,2j} + a^+ \Delta W_{h,\,2i+2,2j}\,.$$

$|\vec{q}_{2i+1,2j}|$ could be used instead of $\mathrm{M}_{2i+1,2j}$. There was no effect on the robustness detectable for either kind of upwind interpolation coefficients.

Instead of Eq. (6.3) an (unweighted) upwind directed prolongation was investigated:

$$\begin{aligned} \Delta W_{h,\,2i+1,2j} \;=\; & 0.5\,(1 + \mathrm{sign}(\mathrm{M}_{2i+1,2j}))\, \Delta W_{h,\,2i,2j} \\ & + 0.5\,(1 - \mathrm{sign}(\mathrm{M}_{2i+1,2j}))\, \Delta W_{h,\,2i+2,2j} \end{aligned} \tag{6.4}$$

with the sign-function resulting in ± 1.0 depending on the sign of the variable. This upwind directed prolongation is schematically shown in Fig. 6.1. As in Fig. 3.7, the thick and dashed lines stand for the coarse and fine grid respectively and the arrows show the possible transfer and weights of coarse grid values to fine grid points. Like the bilinear interpolation this transfer is done by one sweep for each coordinate direction. Again the local velocity could be used instead of the local Mach number.

Calculating fine grid values in an upwinding manner considering only the local flow direction, Eq. (6.4), was not found to stabilize the multigrid method. Therefore no related computational results are presented here.

6.1.1.1.2 Analysis of Upwind Directed Prolongation

The effect of the upwind directed prolongation operator is illustrated in Fig. 6.2 for a one dimensional example: $a^- = 1.0$ and $a^+ = 0.0$ together with $c = 1.0$ are taken to define this operator by means of Eq. (3.108) using Eqs. (3.107) and (3.109).
Assume the coarse grid correction to be represented by a linearly varying function Δ_c on the coarse grid. Transferring Δ_c to the next finer level using the upwind directed prolongation operator results in the piecewise constant function Δ_f on the fine grid.
Implicit residual smoothing as applied to the transferred coarse grid corrections on the finest grid (Sect. 3.3.3.5) transforms Δ_f to a piecewise linear function Δ_s. Thus Δ_s is perturbed compared to Δ_c. This is not the case for the standard prolongation operator in FLOWer, the bilinear interpolation, as demonstrated in Fig. 6.3: The linear variation of Δ_c is transferred correctly to Δ_s.

Only injection as a prolongation operator has been considered in Fourier analysis of multigrid cycles so far. The upwind directed prolongation operator does only affect fine grid points which do not coincide with coarse grid points, i. e. coarse grid corrections are not injected but interpolated to these fine grid points. In order to asses the upwind directed prolongation operator by means of a Fourier analysis at first the Fourier footprint of a 4 level multigrid analysis applied to Eq. (4.1) with a productive source term $S = 1000$ is given in Fig. 6.4. Instead of the injection operator the bilinear interpolation operator as presented in Sect. 3.3.3.4, respectively its Fourier symbol as derived in Sect. 4.4.2.4, has been used for this analysis with $a^- = a^+ = b^- = b^+ = \frac{1}{2}$ and $c = 1.0$. The values of the averaged amplification factors and the absolute value of $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\ (0,0)$ are given in Tab. 6.1 together with the data when using less multigrid levels.
The corresponding Fourier footprint using the upwind directed prolongation operator — $a^- = b^- = 1.0$, $a^+ = b^+ = 0.0$ and $c = 1.0$ — is given in Fig. 6.5. The footprint exhibits some rotation and extension of the area of significantly amplified Fourier modes — $\left|g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\ (\Phi_x, \Phi_y)\right| > 1.2$ — compared to the footprint using bilinear interpolation in Fig. 6.4. Since this rotation affects all Fourier modes having at least one low–frequency component.
In comparison to Tab. 6.1 the same data related to the upwind directed prolongation operator are given in Tab. 6.2. Despite the enlarged area of significantly amplified Fourier modes the averaged data show slightly improved damping properties using the upwind directed prolongation operator for both all and low–frequency Fourier modes.

# multigrid level	$\overline{\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\rvert}_{\mathrm{all}}$	$\overline{\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\rvert}_{\mathrm{low}}$	$\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\,(0,0)\right\rvert$
1	0.40208	0.57503	1.15
2	0.39763	0.52877	1.7248
3	0.40323	0.54987	4.3106
4	0.41649	0.603	21.5416

Table 6.1: Amplification factor averages and its absolute value for $(\Phi_x, \Phi_y) = (0, 0)$ related to the number of multigrid levels considering a productive source term coefficient ($S = 1000$) — multigrid analysis using bilinear interpolation prolongation

# multigrid level	$\overline{\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\rvert}_{\mathrm{all}}$	$\overline{\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\rvert}_{\mathrm{low}}$	$\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\,(0,0)\right\rvert$
1	0.40208	0.57503	1.15
2	0.3926	0.49702	1.7248
3	0.39877	0.52472	4.3106
4	0.41545	0.59173	21.5416

Table 6.2: Amplification factor averages and its absolute value for $(\Phi_x, \Phi_y) = (0, 0)$ related to the number of multigrid levels considering a productive source term coefficient ($S = 1000$) — multigrid analysis using upwind directed prolongation

6.1.1.1.3 Conclusions for Upwind Directed Prolongation

The upwind directed prolongation operator introduced and investigated above has not been found to stabilize the computational method. However, an upwind influence on the prolongation operator is introduced in a different way in the following section.

6.1.1.2 Upwind Weighted Prolongation

After the derivation of the upwind weighted prolongation operator some computational results achieved with it are discussed. Before concluding this section this operator is analyzed by means of a one dimensional example.

6.1.1.2.1 Derivation of Upwind Weighted Prolongation

This nonstabilizing behaviour of the unweighted upwind directed prolongation is the reason why weights have been considered which are in general less than one. This upwind weighted prolongation operator is presented in Fig. 6.6. The prolongation operator, Eq. (6.4), has been changed to

$$\begin{aligned} \Delta W_{h,\,2i+1,2j} &= 0.5\,(\mathrm{M}_{2i+1,2j} + |\mathrm{M}_{2i+1,2j}|)\,\Delta W_{h,\,2i,2j} \\ &\quad + 0.5\,(\mathrm{M}_{2i+1,2j} - |\mathrm{M}_{2i+1,2j}|)\,\Delta W_{h,\,2i+2,2j}\,. \end{aligned} \tag{6.5}$$

Thus no interpolation is performed when calculating fine grid values as interpolation means summing up several weighted values, with weights summing to one.

6.1.1.2.2 Computational Results with Upwind Weighted Prolongation

In Figs. 6.7 – 6.9 the convergence of the density residual for the RAE 2822 test case as well as the k and ω residuals using the upwind weighted prolongation operator for the turbulence equations are plotted. Convergence behaviour has changed in two ways compared to the application of the standard prolongation (Figs. 5.17 – 5.19):
For the cases reaching some level of convergence — at Re $= 6.5 \cdot 10^6$ and at Re $= 20 \cdot 10^6$ — convergence is slowed down for all residuals by a factor of about 1.5. However, at Re $= 60 \cdot 10^6$ all residuals start to converge. Compared to the application of the bilinear prolongation operator three additional orders of magnitude in convergence are reached before all residuals turn to diverge after about 4000 iterations. Thus some stabilization is introduced to the multigrid method for the turbulence equations by this new prolongation operator.

6.1.1.2.3 Analysis of Upwind Weighted Prolongation

This stabilizing effect is illustrated in Fig. 6.10 for the same one dimensional example as above: The weighted upwind prolongation operator used here is represented by Eq. (3.108) using Eqs. (3.107) and (3.109) with $a^- = 0.01$, $a^+ = 0.0$ and $c = 1.0$.
Application of this operator to transfer the linear coarse grid function Δ_c results in an oscillating fine grid function Δ_f. If this function is subject to an implicit smoothing with a constant coefficient $\varepsilon = 0.2$ it results in the function Δ_s which is still oscillating but with reduced amplitudes.

# multigrid level	$\overline{\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\rvert}_{\mathrm{all}}$	$\overline{\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\rvert}_{\mathrm{low}}$	$\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\,(0,0)\right\rvert$
1	0.40208	0.57503	1.15
2	0.40205	0.57489	1.1505
3	0.40205	0.57489	1.1505
4	0.40205	0.57489	1.1505

Table 6.3: Amplification factor averages and its absolute value for $(\Phi_x, \Phi_y) = (0, 0)$ related to the number of multigrid levels considering a productive source term coefficient ($S = 1000$) — multigrid analysis using upwind weighted prolongation (velocity as weighting factor)

A Fourier analysis as in Sect. 6.1.1.1.2 using the upwind weighted prolongation operator leads to the Fourier footprint given in Fig. 6.11. It shows nearly no differences to the Fourier footprint using on multigrid level only, Fig. 5.51. The same result can be obtained from Tab. 6.3. Any multigrid effect is eliminated due to the weighting incorporated in the prolongation operator.

6.1.1.2.4 Conclusions for Upwind Weighted Prolongation

Interpreting Δ_c again as coarse grid corrections this example shows that the application of the upwind weighted prolongation operator yields a reduction of the coarse grid corrections transferred to the finest grid. Thus the influence of multigrid on the convergence of the turbulence equations is reduced.

The following section will show that this underrelaxation of the coarse grid corrections is the reason for the gain in stability of the multigrid method.

6.1.2 Relaxed Linear Prolongation

Based on the experiences from the previous section a relaxed linear prolongation operator for the turbulence equations is derived subsequently. After a brief presentation of computational results this operator is thoroughly analyzed. Conclusions bring this section to a close.

6.1.2.1 Derivation of Relaxed Linear Prolongation

For the upwind weighted prolongation the calculated fine grid values are underrelaxed but less smooth compared to those fine grid values coming from linear interpolation. Therefore the idea arose not to replace the linear interpolation but to relax the injection of coarse grid values to fine grid points, i. e. to replace the weights 1.0 in Fig. 3.7 by some other relaxation coefficients $c_{i,j}$:

$$\Delta W_{h,\,2i,2j} = c_{i,j} \Delta W_{2h,\,i,j}\,. \tag{6.6}$$

This is illustrated in Fig. 6.12. After such a relaxed injection a standard linear interpolation as denoted in Eq. (3.108) and Eq. (3.109), respectively, is performed. Venkatakrishnan [127] mentions a similar treatment for the Spalart–Allmaras turbulence model with constant relaxation coefficients equal to 0.5. Using the same constant relaxation coefficient did not improve the robustness of the multigrid method applied to the k–ω turbulence model for the test cases used within this work.

6.1.2.2 Computational Results with Relaxed Linear Prolongation

The most efficient relaxation coefficient $c_{i,j}$ was found to be the local Mach number. It should be used in a normalized and limited version, i. e. the local value is normalized by means of the freestream Mach number and afterwards limited from above by 1.0.
The application of this transfer operator improves the robustness of the multigrid method significantly as shown in Fig. 6.13: The density residual at $\mathrm{Re} = 60 \cdot 10^6$ converges about eight orders of magnitude. Furthermore, the slow down in convergence at $\mathrm{Re} = 6.5 \cdot 10^6$ and $\mathrm{Re} = 20 \cdot 10^6$ is negligible compared to the reference calculations in Fig. 5.17. Analogue results for the turbulence residuals are given in Fig. 6.14 and Fig. 6.15. However, at $\mathrm{Re} = 100 \cdot 10^6$ convergence is not obtained by this approach either.
It follows that the convergence slow down introduced by the upwind weighted prolongation operator (Figs. 6.7 – 6.9) can be put down to the perturbations caused by its upwind character (Fig. 6.10).

6.1.2.3 Analysis of Relaxed Linear Prolongation

In Fig. 6.16 the relaxed linear interpolation is demonstrated for the same example as the standard prolongation in Fig. 6.3. For this example $c = 0.01$ has been chosen. The transferred and smoothed result Δ_s still varies linearly but its slope has been multiplied by c. Note that the relaxation coefficients need not be constant. So relax–injected constant functions will take the slope of their relaxation coefficients.

A Fourier analysis of multigrid using the relaxed linear prolongation operator has also been performed. In Sect. 4.4.2.4 it has been noted that the Fourier symbol of the standard, i. e. unrelaxed injection equals 1. As the injection is relaxed here its Fourier symbol is multiplied by the same relaxation coefficient c. Thus the Fourier symbol of this prolongation operator reads

$$\mathcal{P}_{2h}^{h} = c_{i,j} \,. \tag{6.7}$$

As stability problems are increased by multigrid treatment of productive source terms the Fourier analysis of this new prolongation operator has been performed for the same test case as in Sect. 5.2.3.2. The relaxation coefficient $c_{i,j}$ has been chosen to be 0.1 for this analysis. However, it was not possible to obtain computational results for the RAE 2822 airfoil at any Reynolds number using this constant relaxation coefficient. The Fourier footprint of a two level method acting on a two dimensional convection diffusion equation with a productive source term — $S = 1000$ — is shown in Fig. 6.17. Comparing this figure to its reference plot in Fig. 5.53 and to the Fourier footprint of the corresponding singlegrid method in Fig. 5.51 clarifies the significant reduction of the coarse grid's influence further:
Fig. 6.17 differs from Fig. 5.51 in two regions: Around $\Phi_x = 0.0$ the damping properties are slightly improved for $|\Phi_y| \in [\pi/2, \pi]$. It is much more important that the amplification of modes is only slightly increased around $(\Phi_x, \Phi_y) = (0, 0)$. This becomes more obvious comparing Tab. 6.4 to Tab. 5.7.

# multigrid level	$\overline{\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\rvert}_{\mathrm{all}}$	$\overline{\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\rvert}_{\mathrm{low}}$	$\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\ (0,0)\right\rvert$
1	0.40208	0.57503	1.15
2	0.39542	0.55921	1.2074
3	0.3953	0.5588	1.2333
4	0.3953	0.55879	1.2505

Table 6.4: Amplification factor averages and its absolute value for $(\Phi_x, \Phi_y) = (0, 0)$ related to the number of multigrid levels considering a productive source term coefficient ($S = 1000$) and relaxed linear prolongation ($c = 0.01$)

Adding a third and a fourth level for the Fourier analysis yields only very small changes of the Fourier footprint as shown in Fig. 6.18 and Fig. 6.19 respectively. The same result can be obtained from Tab. 6.4. The overall averaged damping factor $\overline{\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\rvert}_{\mathrm{all}}$ is identical — within the accuracy used for the output of these results — for the three level resp. four level method . The difference to the two level method is about 10^{-4}. Nearly the same holds for the averaged amplification factor of the low–frequency modes, $\overline{\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\right\rvert}_{\mathrm{low}}$. Only the amplification factor $\left\lvert g^{(q=5)}_{\mathrm{pi-src,\,IRS}}\ (0,0)\right\rvert$ still grows. However, its growth is negligible compared to the case of unrelaxed injection as given in Tab. 5.7.
Coarse grid corrections from the third grid level influence the fine grid by only 1% of their original value due to the relaxed linear prolongation operator with $c = 0.1$. The fourth grid's influence is reduced by a factor of $1/1000$. Thus the multigrid treatment of the turbulence equations is nearly switched off by means of this new prolongation operator.

6.1.2.4 Conclusions for Relaxed Linear Prolongation

The relaxed linear prolongation is based on the idea to reduce the influence of the multigrid method on the numerical treatment of the turbulence equations.
A serious problem is to find suitable relaxation coefficients as they should vary reciprocally to the production of turbulence: In regions of turbulence production these coefficients should be close to 0.0. In any other region they are to be nearly 1.0 in order to keep multigrid effective. Although the local Mach number tends to zero in the vicinity of no–slip walls it is not related to turbulence production in any way.
The sensor and the scaling factor suggested by Gerlinger et al. [40] as given in Eqs. (6.2) and (6.1) fulfill this requirement of reciprocal variation. Their case dependence is still a crucial point. Additionally the question arose what maximum of which source terms should be taken. As the FLOWer code is capable of multiblock topologies one has to distinguish between block local and global maxima. This approach has not been investigated within this work since first very promising results of an implicit treatment of the turbulence equations — as described in Sect. 6.3 — had been obtained around the date, when the work of Gerlinger et al was published. Moreover, the results from Fourier analysis shown in Sect. 6.1.2.3 do not indicate a significantly improved convergence for different scaling factors.

A strict consequence of the results presented in this section is to omit the prolongation of coarse grid corrections at fine grid points with a productive source term. This is discussed in the following section.

6.1.3 Multigrid Update Omission Governed by Source Terms

The derivation of the treatment of coarse grid corrections presented in this section is followed by some computational results. Conclusions are drawn at the end of this section.

6.1.3.1 Derivation of Multigrid Update Omission Governed by Source Terms

Mavriplis and Martinelli [80] proposed to set those coarse grid corrections to zero which would yield negative values — i. e. values below the chosen limiter — of the turbulence quantities on the finest grid. These coarse grid corrections are directly related to destructive turbulence source terms. Instead of Eq. (3.125c) the coarse grid corrections are modified as follows:

$$\Delta_H W = I_{2H}^H \circ \Delta W_{2H} \tag{6.8a}$$

$$W_{H;\,\mathrm{pseudo}} = W_H + \Delta_H W \tag{6.8b}$$

$$W_{H;\,\mathrm{pseudo;\,lim}} = \begin{cases} W_H & \text{IF} \quad W_{H;\,\mathrm{pseudo}} \leq C_{W;\mathrm{lim}} \cdot W_\infty \\ W_{H;\,\mathrm{pseudo}} & \text{ELSE} \end{cases} \tag{6.8c}$$

$$\Delta_{H;\,\mathrm{lim}} W = W_{H;\,\mathrm{pseudo;\,lim}} - W_H \tag{6.8d}$$

This approach is based on the omission of multigrid updates at fine grid points with a destructive source term.

On the other hand productive source terms have been shown in Sect. 5.2.3 to yield robustness problems with a multigrid method. These problems can be reduced significantly if the coarse grid corrections are underrelaxed as derived in Sect. 6.1.2. Thinking out the omission of multigrid updates at points with productive source terms is the consequence.

Combining both approaches consequently results in switching off multigrid for the turbulence equations.

6.1.3.2 Computational Results without Multigrid Updates for Turbulence Equations

In Fig. 6.20 the convergence of the density residual is shown which is still sped up by a four level W–cycle. Comparing to the corresponding calculations with standard multigrid treatment of the turbulence equations (Fig. 5.17) the convergence at Re $= 6.5 \cdot 10^6$ is slowed down by a factor of about 1.5 but reaches an additional order of convergence. For this Reynolds number it has been found to be necessary to increase the CFL number for time integration of the turbulence equations, `CFLTU`. This parameter has been introduced to the FLOWer code to enable the adjustment of the time step to a modified time integration approach for the turbulence equations. For the computation at Re $= 6.5 \cdot 10^6$ has been set to `CFLTU` $= 10$. Otherwise — in case `CFLTU` $=$ `CFL` — the computation diverged. This is obviously due to a decoupling in time of the RANS and the turbulence equations.
However, much more important is the change in the convergence behaviour of the density residual at Re $= 60 \cdot 10^6$. Applying the same four level W–cycle to the turbulence equations as to the RANS equations did not result in a converged solution (Fig. 5.23). Switching off multigrid for the turbulence equations but keeping the four level W–cycle for the mean equations results in a convergence of the density residual better than applying a four level V–cycle to all equations as shown in Fig. 5.24.
Figs. 6.21 and 6.22 present the convergence of the k resp. ω residual. Both do not inherit any multigrid treatment but converge to machine zero for all Reynolds numbers except Re $= 100 \cdot 10^6$. Machine accuracy has not been reached by the turbulence residuals in any case if the turbulence equations are "sped up" by the application of multigrid.

A remark on the resulting aerodynamic coefficients at Re $= 6.5 \cdot 10^6$ is necessary: Using singlegrid for the k–ω turbulence model but applying a 4 level W–cycle to the RANS equations results in identical lift and drag coefficients as given in Tab. 5.1 for the singlegrid computation. There both RANS and k–ω equations have been treated by singlegrid only.

6.1.3.3 Conclusions for Multigrid Update Omission Governed by Source Terms

Robustness of the overall method has been improved significantly by treating turbulence equations with singlegrid only. At the same time the multigrid treatment of the RANS equations has not been affected. However, the convergence of the density residual is slowed down.

6.1.4 Summary of Improvements to Multigrid

All multigrid modifications investigated are based on the idea to reduce the influence of the multigrid method on the numerical treatment of the turbulence equations. In effect the multigrid method has been switched off for the turbulence equations, i e. while the RANS equations are still subject to multigrid the turbulence equations are treated with singlegrid only. Robustness of the complete method has been increased significantly by this approach.

Furthermore, the influence of the lower bounds for k and ω on the level of convergence reached by the turbulence residuals has been eliminated. As this influence has been exerted by the forcing function as discussed in Sect. 5.1.3.3.3 this observation is reasonable.
On the other hand the level of convergence reached by the density residual does not change noticeably whether multigrid is applied to the turbulence equations or not. This will be discussed in the following section.

6.2 Improvements to the Basic Scheme

This section is dedicated to present the modifications of the basic scheme that have been implemented in the FLOWer code. In Sect. 6.2.1 the coupling between the k–ω turbulence model and the RANS equations is discussed. A modified time step for the turbulence equations is investigated in Sect. 6.2.2. In Sect. 6.2.3 the linearization of the point implicitly treated source term performed before the application of the Runge–Kutta method is discussed. This alternative approach has been noted in Sect. 3.2.2.2. This section is summarized in Sect. 6.2.4.

For the first two modifications no results from Fourier analysis are presented. The coupling strategy can not be taken into account by Fourier analysis as used here, since no system of equations is investigated. Furthermore, the Fourier analysis of the model problem, Eq. (4.1), is not capable of the modified time step as no speed of sound is defined within the Fourier analysis which is — as shown below — essential for this modification.

6.2.1 Modified Coupling of RANS and Turbulence Equations

In Sect. 5.1.3.3.3 the influence of the lower bounds for the k–ω turbulence model on its convergence behaviour has been investigated. The results showed an influence of these bounds on the convergence of the density residual, too. It has been found that this influence on the RANS equations is introduced via the contributions of k to the energy E, Eq. (2.28), i. e. to the viscous energy fluxes, Eq. (2.35).
Two coupling strategies between the RANS and the turbulence equations are investigated in this section. On the one hand a physically consistent coupling is presented in Sect. 6.2.1.1. It is called "maximized" coupling as all data provided by the k–ω turbulence model are used in all terms of the RANS equations wherever physically correct. The data relevant for the RANS equations is the turbulent kinetic energy k besides

the eddy viscosity μ_{tu}. On the other hand a "minimized" coupling is discussed in Sect. 6.2.1.2. By this approach only the eddy viscosity as turbulence model related data is used by the RANS equations.

6.2.1.1 Maximized Coupling

A brief derivation of this coupling strategy is given at first. Secondly computational results are discussed before conclusions finish this section.

6.2.1.1.1 Derivation of Maximized Coupling

In Sect. 2.2 the neglection of $-2/3\rho k$ on the main diagonal of the Reynolds stress tensor has been noted. Firstly it has been assumed that this neglection may be the reason for the influence of the turbulence quantities' bounds, as the neglection is physically inconsistent. Thus the neglected contributions of k have been taken into account again.

6.2.1.1.2 Computational Results with Maximized Coupling

The influence of the bounds for the turbulence quantities on the convergence behaviour of the density residual has not been removed by this approach as shown in Fig. 6.23. In contrast to the expected effect the convergence has been deteriorated for all residuals as found in Figs. 6.24 – 6.25 compared to Figs. 5.17 – 5.19.
Slightly improved convergence behaviour is obtained if only singlegrid is applied to the turbulence equations appropriate to the previous section as shown in Figs. 6.26 – 6.28. Especially the turbulence residuals — Fig. 6.27 and Fig. 6.28 — presumably converge for Re $= 6.5 \cdot 10^6$. Furthermore, all residuals exhibit an increased robustness since at Re $= 60 \cdot 10^6$ the residuals do not wiggle any longer.

# multigrid level	lift	drag
experiment	0.803	0.0168
SG	0.802724	0.0183064
W4	0.803084	0.0183225

Table 6.5: Aerodynamic coefficients for lift and drag related to number of multigrid levels used for k–ω turbulence model with maximized coupling (RANS : 4 multigrid level, W-cycle); RAE 2822 airfoil at Re $= 6.5 \cdot 10^6$

According to the aerodynamic coefficients for the RAE 2822 airfoil at Re $= 6.5 \cdot 10^6$ the maximized coupling eliminates the difference to the experimental results. Comparing Tab. 6.5 to Tab. 5.1 shows an increase of more than 4 lift counts for the computed lift coefficient. Thus the lift coefficient computed by means of the maximized coupling approach meets the experimental data also given in Tab. 6.5 nearly independent from the number of multigrid levels used for the k–ω turbulence model. The influence of the number of multigrid levels applied to the k–ω turbulence model does not change with the coupling strategy: switching from singlegrid to a 4 level W–cycle increases the lift coefficient by little more than 0.35 lift counts.

The drag coefficient is increased by coupling the k–ω turbulence model in a maximum way to the RANS equations but the difference to the experimental data is also increased. However, the difference the experimental data is about 15 drag counts. Less than 2 drag counts have been added due to the coupling strategy. However, the difference in the drag coefficient between the computed values and the experimental one is comparable to the difference given by Rudnik [105] for this turbulence model.

6.2.1.1.3 Conclusions for Maximized Coupling

Comparing Figs. 6.26 – 6.28 to results based on the original coupling strategy, Figs. 6.20 – 6.22, it is obvious that this maximized coupling is not the way to proceed: The convergence levels reached are worse and the convergence behaviour of the turbulence residuals is less than satisfactory.

6.2.1.2 Minimized Coupling

A detailed derivation of this coupling approach is presented before computational results are discussed. Again conclusions bring this section to a close.

6.2.1.2.1 Derivation of Minimized Coupling

In the literature no comments have been found on the coupling between the k–ω turbulence model and the RANS equations. In the user's manual of CFL3D [60], a flow solver from the NASA Langley Research Center, and in the technical documentation of elsA [91], the flow solver from ONERA, it has been found, that in both codes k does not contribute to the RANS equations in any way. They argue that k is significantly smaller then $e + (u^2 + v^2 + w^2)/2$. Thus neglection of k in the viscous energy fluxes and on the main diagonal of the Reynolds stress tensor is expected not to change the solution.
In addition to these hints from CFL3D and elsA the neglection of k in the RANS equations yields a common coupling strategy for any turbulence model. All turbulence models provide the eddy viscosity, thus the eddy viscosity is the only coupling quantity.

According to this approach a new option has been implemented in the FLOWer code, i. e. the user can choose the RANS equations as introduced in Sect. 2.1.1 although using a k–ω turbulence model:
Neglecting the contributions of k to the energy E, i. e. using

$$E = e + \frac{u^2 + v^2 + w^2}{2} \tag{2.7}$$

instead of

$$E = e + \frac{u^2 + v^2 + w^2}{2} + k \tag{2.28}$$

forces the neglection of k in the viscous energy fluxes, too, i. e. using

$$
\begin{aligned}
\Psi_x &= u\sigma_{xx} + v\sigma_{xy} + w\sigma_{xz} + \mathtt{k}\frac{\partial T}{\partial x}\,, \\
\Psi_y &= u\sigma_{xy} + v\sigma_{yy} + w\sigma_{yz} + \mathtt{k}\frac{\partial T}{\partial y}\,, \\
\Psi_z &= u\sigma_{xz} + v\sigma_{yz} + w\sigma_{zz} + \mathtt{k}\frac{\partial T}{\partial z}\,.
\end{aligned}
\tag{2.19}
$$

instead of

$$
\begin{aligned}
\Psi_x &= u\sigma_{xx} + v\sigma_{xy} + w\sigma_{xz} + \mu_k\frac{\partial k}{\partial x} + \mathtt{k}\frac{\partial T}{\partial x}\,, \\
\Psi_y &= u\sigma_{xy} + v\sigma_{yy} + w\sigma_{yz} + \mu_k\frac{\partial k}{\partial y} + \mathtt{k}\frac{\partial T}{\partial y}\,, \\
\Psi_z &= u\sigma_{xz} + v\sigma_{yz} + w\sigma_{zz} + \mu_k\frac{\partial k}{\partial z} + \mathtt{k}\frac{\partial T}{\partial z}\,.
\end{aligned}
\tag{2.35}
$$

Furthermore the contributions of k to the main diagonal of the Reynolds stress tensor are neglected, i. e. use

$$
\begin{aligned}
\sigma_{xx} &= 2\mu\frac{\partial u}{\partial x} - \frac{2}{3}\mu\,\mathrm{div}\,(\vec{q})\,, \\
\sigma_{yy} &= 2\mu\frac{\partial v}{\partial y} - \frac{2}{3}\mu\,\mathrm{div}\,(\vec{q})\,, \\
\sigma_{zz} &= 2\mu\frac{\partial w}{\partial z} - \frac{2}{3}\mu\,\mathrm{div}\,(\vec{q})
\end{aligned}
\tag{2.15}
$$

instead of

$$
\begin{aligned}
\sigma_{xx} &= 2\mu\frac{\partial u}{\partial x} - \frac{2}{3}\mu\,\mathrm{div}\,(\vec{q}) - \frac{2}{3}\rho k\,, \\
\sigma_{yy} &= 2\mu\frac{\partial v}{\partial y} - \frac{2}{3}\mu\,\mathrm{div}\,(\vec{q}) - \frac{2}{3}\rho k\,, \\
\sigma_{zz} &= 2\mu\frac{\partial w}{\partial z} - \frac{2}{3}\mu\,\mathrm{div}\,(\vec{q}) - \frac{2}{3}\rho k\,.
\end{aligned}
\tag{2.30}
$$

6.2.1.2.2 Computational Results with Minimized Coupling

Results obtained by this minimized coupling are presented in Figs. 6.29 – 6.31. Fig. 6.29 shows that this coupling ensures a convergence of the density residual to machine accuracy. From these results together with the formulae presented above it can be concluded that the convergence behaviour of the density residual becomes independent of the value of RTULF, the lower bound of the turbulence quantities.

However, there is no effect on the turbulence residuals as shown in Fig. 6.30 and Fig. 6.31 compared to Fig. 5.18 and Fig. 5.19 respectively. Also no increased robustness is gained: In Figs. 6.29 – 6.31 no change in the convergence behaviour of any residual at Re $= 60 \cdot 10^6$ nor at Re $= 100 \cdot 10^6$ can be observed.
Based on the results presented in Sect. 6.1 the multigrid treatment of the turbulence equations has been switched off. This results in convergence as plotted in Figs. 6.32 – 6.34. Again the convergence behaviour of the turbulence residuals is independent of the coupling strategy as can be seen in Fig. 6.33 compared to Fig. 6.21 resp. in Fig. 6.34 compared to Fig. 6.22. For the density residual two effects combine: The convergence independent of the turbulence quantities' bounds arose from the minimized coupling between RANS and turbulence equations. This can be seen by comparing Fig. 6.32 to Fig. 6.20. On the other hand robustness is increased due to the singlegrid treatment of the turbulence equations, i. e. at Re $= 60 \cdot 10^6$ convergence of all residuals to machine accuracy is obtainable.
Like in Sect. 6.1.3.2 for the computation at Re $= 6.5 \cdot 10^6$ the CFL number for time integration of the turbulence equations, `CFLTU`, has been increased to `CFLTU` $= 10$ in order to obtain convergence.

As announced in Sect. 5.1.4.2 the coupling strategy has an effect on the forcing function. The same calculations for the decay of turbulence as in Sect. 5.1.4.2 have been performed. This time k does not contribute to the RANS equations, thus the forcing function of the mean equations is not influenced by k. For the density residual in Fig. 6.35 a different plot style has been used to show that the residual equals zero for all iterations, independent of the number of multigrid levels different from Fig. 5.43.

# multigrid level	lift	drag
SG	0.798328	0.0181448
W4	0.798694	0.0181608

Table 6.6: Aerodynamic coefficients for lift and drag related to number of multigrid levels used for k–ω turbulence model with minimized coupling (RANS : 4 multigrid level, W-cycle); RAE 2822 airfoil at Re $= 6.5 \cdot 10^6$

According to the aerodynamic coefficients for the RAE 2822 airfoil at Re $= 6.5 \cdot 10^6$ the minimized coupling changes the computed results negligibly. Comparing Tab. 6.6 to Tab. 5.1 shows that the lift coefficient is reduced by about 0.03 lift counts due to the minimized coupling instead of the default coupling. The drag coefficient computed by means of the minimized coupling approach is about 0.01 drag counts less than the corresponding ones given in Tab. 5.1. Again it can be observed that the influence of the number of multigrid levels applied to the k–ω turbulence model does not change with the coupling strategy.

6.2.1.2.3 Conclusions for Minimized Coupling

It has been found that the minimized coupling strategy decouples the mean equations from the turbulence equations as much as possible. This coupling via the eddy viscosity only lead to an improved convergence behaviour of the density residual, as it is no longer influenced by the lower bound of turbulence quantities.

Combining this approach with the singlegrid treatment of the turbulence equations yielded convergence to machine zero for all residuals up to $\mathrm{Re} = 60 \cdot 10^6$.

6.2.1.3 Conclusions for Modified Coupling of RANS and Turbulence Equations

Two strategies for the coupling of the turbulence equations to the mean equations, which differ from the standard coupling in the FLOWer code, have been presented and discussed.

Although it is physically more correct, the maximized coupling did not improve the convergence behaviour of any residual. However, the computed lift coefficient at $\mathrm{Re} = 6.5 \cdot 10^6$ meets the experimental one nearly independent of the number of multigrid levels applied to the turbulence equations.
The weaker coupling via eddy viscosity only yielded convergence of the density residual independent of the lower bound of the turbulence quantities. Combined with the singlegrid treatment for the turbulence equations all residuals converge to machine zero at all Reynolds numbers except for $\mathrm{Re} = 100 \cdot 10^6$.
The results obtained by the different coupling strategies differ less than 5 lift counts and less than 2 drag counts, respectively. This has to be compared to the difference of about 7 lift counts which may result from a different turbulence model, the algebraic turbulence model by Baldwin and Lomax [4].

6.2.2 Modified Time Step for Turbulence Equations

Firstly a separate time step for the turbulence equations is derived. Computational results are discussed before some conclusions close this section.

6.2.2.1 Derivation of Modified Time Step for Turbulence Equations

Originally the time step for the time integration of the turbulence equations has been taken the same as for the RANS equations:

$$\Delta t_{i,j} = \frac{\widetilde{\mathcal{V}}_{i,j}^2}{\widetilde{\mathcal{V}}_{i,j} \cdot (\lambda_\xi + \lambda_\eta) + \mathtt{DTVI} \cdot \frac{\mu_{l;i,j} + \mu_{tu;i,j}}{\rho_{i,j}} (\mathcal{C}_\xi \cdot \mathcal{C}_\xi + \mathcal{C}_\eta \cdot \mathcal{C}_\eta)} \tag{3.73}$$

Thus the time step for all equations is based on the sum of the maximum eigenvalues, i. e. the characteristic speeds λ_*, derived for the RANS equations. λ_* is calculated according to Eq. (3.74) and represents the maximum speed at which information is transported through the fluid. Since acoustic waves carry information, the speed of sound has to be added to the velocity of the moving fluid. On the other hand the turbulence equations do not represent any acoustic effects. Thus the corresponding characteristic speed $\lambda_{tu;*}$ is calculated as [86]

$$\lambda_{tu;*} = |\vec{q} \bullet (\mathcal{C}\vec{n})_*| \ . \tag{6.9}$$

Since $\lambda_{tu;*} \ll \lambda_*$ for $q_{\vec{n}} \to 0$, using the same time step as for the RANS equations results in a very conservative time step restriction for the turbulence equations. On the other hand it was found that the use of

$$\Delta t_{tu;i,j} = \frac{\widetilde{\mathcal{V}}_{i,j}^2}{\widetilde{\mathcal{V}}_{i,j} \cdot (\lambda_{tu;\xi} + \lambda_{tu;\eta}) + \texttt{DTVI} \cdot \dfrac{\mu_{l;i,j} + \mu_{tu;i,j}}{\rho_{i,j}} (\mathcal{C}_\xi \cdot \mathcal{C}_\xi + \mathcal{C}_\eta \cdot \mathcal{C}_\eta)} \tag{6.10}$$

with $\lambda_{tu;*}$ calculated as above leads to an unstable convergence behaviour. This can be traced to very large $\Delta t_{tu;i,j}$ in case of $\vec{q}_{i,j} \to 0$, which is the case in the vicinity of solid wall boundaries. Therefore R. Emunds [36] suggested to modify the turbulent eigenvalues as

$$\lambda_{tu;*} = |\vec{q} \bullet (\mathcal{C}\vec{n})_*| + \texttt{TUSOUNDF} \cdot a_{i,j} \cdot \mathcal{C}_* \tag{6.11}$$

with $\texttt{TUSOUNDF} = 0.15$, which was found to yield stability.

6.2.2.2 Computational Results with Modified Time Step for Turbulence Equations

If this modified time step is used within a four level W–cycle one has to expect that the scheme gets less robust due to the larger time step for the turbulence equations. This has been confirmed by calculations: At no Reynolds number a solution has been reached.
In Figs. 6.36 – 6.38 a singlegrid approach together with the enlarged time step is applied to the turbulence equations. At all Reynolds numbers $\texttt{CFLTU} = 7.5$ has been used together with $\texttt{CFL} = 7.5$ different from the calculation at $\text{Re} = 6.5 \cdot 10^6$ discussed in Sect. 6.1.3.2.
All residuals show a shortened period of wiggling before they start to converge compared to Figs. 6.20 – 6.22. The convergence behaviour of the turbulence equations at $\text{Re} = 6.5 \cdot 10^6$ is difficult to compare since using the standard time step CFLTU has been increased compared to the computations with the enlarged time step for the turbulence equations: For the ω residual, Fig. 6.38 compared to Fig. 6.22, the enlarged time step has nearly the same effect as the increased CFL number CFLTU. A speed–up of about 10% can be observed at $\text{Re} = 20 \cdot 10^6$. At this Reynolds number for the standard and the modified time step $\texttt{CFLTU} = 7.5$ has been taken. The k residual at $\text{Re} = 6.5 \cdot 10^6$ reaches machine zero about 2000 iterations earlier if the CFL number CFLTU is increased using the standard time step compared to the modified time step for the turbulence equations in combination with $\texttt{CFLTU} = 7.5$.

As a next step the coupling between the mean and the turbulence equations as described in Sect. 6.2.1 has been minimized in addition to both the enlarged time step and the singlegrid approach for the turbulence equations. Corresponding results are shown in Figs. 6.39 – 6.41. The k residual as well as the ω residual reach machine zero without considerable changes in convergence compared to results with standard coupling as in Fig. 6.37 and Fig. 6.38, respectively. However, the density residual (Fig. 6.39) needs more iterations at $\text{Re} = 6.5 \cdot 10^6$ compared to Fig. 6.32. At higher Reynolds numbers no further differences are visible.

6.2.2.3 Conclusions from Modified Time Step for Turbulence Equations

An enlarged time step based on a corrected hyperbolic eigenvalue of the turbulence equations has been introduced. This eigenvalue has not been set to its mathematically correct value for stability reasons.
This enlarged time step for the turbulence equations yielded a speed–up not only for the convergence of the k and ω residuals but for the convergence of the density residual, too.
Highest efficiency has been achieved if all optimizations found are combined: Both the enlarged time step and the singlegrid treatment acting only on the turbulence equations whereas the minimized coupling directly improves convergence of density. Additionally an increased robustness at higher Reynolds numbers (up to $\mathrm{Re} = 60 \cdot 10^6$) is gained.

6.2.3 Alternative Linearization of Point Implicitly Treated Source Term

Firstly the alternative linearization of the point implicitly treated source term as noted in Sect. 3.2.2.2 is derived. Computational results are discussed before some conclusions close this section.

6.2.3.1 Derivation of Alternative Linearization

As noted in Sect. 3.2.2.2 the point implicitly treated source term in

$$W^{n+1} = W^n + \tau \left(Z_c \circ \; + Z_v \circ \; + Z_S \circ \mathcal{T} \circ \right) W^n \tag{3.63}$$

may be linearized before or after the Runge–Kutta method is applied to this equation. While the linearization of Eq. (3.63) after the application of the time stepping scheme has been discussed in Sect. 3.2.2.2 here the alternative linearization is presented.

Linearization of Eq. (3.63) with respect to the old time level before the Runge–Kutta method is applied leads to

$$W^{n+1} = W^n + \frac{\tau}{1 - \tau \dfrac{\partial \left(Z_S \circ W^n \right)}{\partial W}} \left(Z_c \circ \; + Z_v \circ \; + Z_S \circ \right) W^n \,. \tag{6.12}$$

Applying the Runge–Kutta scheme, Eq. (3.54), to Eq. (6.12) the qth stage reads

$$W^{(q)} = W^{(0)} + \alpha_q^{\text{alt. pi}-\text{src}} \left(Z_c \circ W^{(q-1)} + Z_v \circ W^{(0)} + Z_S \circ W^{(0)} \right) \tag{6.13}$$

with

$$\alpha_q^{\text{alt. pi}-\text{src}} = \frac{\alpha_q}{1 - \tau \dfrac{\partial \left(Z_S \circ W^{(0)} \right)}{\partial W}} \,. \tag{6.14}$$

Thus the two approaches (Eq. (3.66) and Eq. (6.13)) of applying the Runge–Kutta scheme to the point implicitly treated source term, Eq. (3.63), only differ in the factor α_q in the denominator of the modified stage coefficients (Eq. (3.67) and Eq. (6.14)) on the qth stage.

6.2.3.2 Analysis of Alternative Linearization

In the same way as Eq. (4.48) is derived in Sect. 4.4.1 the Fourier symbol of the Runge–Kutta method applying the alternative linearization to the source term is gained. It reads

$$\begin{aligned} f^{(q)}_{\text{alt. pi}-\text{src, IRS}} \;=\; \mathcal{B}\Bigg[& \frac{\alpha_q}{1-\tau\min\left(0,\mathcal{Z}_S\right)} \\ & + \frac{\alpha_q}{1-\tau\min\left(0,\mathcal{Z}_S\right)}\,\frac{\alpha_{q-1}}{1-\tau\min\left(0,\mathcal{Z}_S\right)}\,\tau\mathcal{Z}_c\mathcal{B} \\ & + \ldots + \prod_{r=1}^{q}\frac{\alpha_r}{1-\tau\min\left(0,\mathcal{Z}_S\right)}\left(\tau\mathcal{Z}_c\mathcal{B}\right)^{q-1}\Bigg]\,. \end{aligned} \tag{6.15}$$

Inserting $f^{(q)}_{\text{alt. pi}-\text{src, IRS}}$ instead of $f^{(q)}_{\text{pi}-\text{src, IRS}}$ in the definition of $g^{(q=5)}_{\text{pi}-\text{src, IRS}}$ yields

$$g^{(q=5)}_{\text{alt. pi}-\text{src, IRS}} = 1 + \tau f^{(q=5)}_{\text{alt. pi}-\text{src, IRS}}\,\mathcal{Z}\,. \tag{6.16}$$

A Fourier analysis with a destructive source term $S = -1000$ has been performed. The number of multigrid levels has been varied. The analysis based on the amplification factor $g^{(q=5)}_{\text{alt. pi}-\text{src, IRS}}$ resulted in the Fourier footprints presented in Figs. 6.42 – 6.45. These plots are to be compared to Fig. 5.46 and Figs. 5.48 – 5.50.
Increasing the number of multigrid levels the alternative linearization of the point implicitly treated source term shows more and more significantly better damping properties for the low–frequency modes compared to the standard linearization. This surpassing is also obvious from Tab. 6.7 in comparison to Tab. 5.5 on page 85.

# multigrid level	$\overline{\left\lvert g^{(q=5)}_{\text{alt. pi}-\text{src, IRS}}\right\rvert}_{\text{all}}$	$\overline{\left\lvert g^{(q=5)}_{\text{alt. pi}-\text{src, IRS}}\right\rvert}_{\text{low}}$	$\left\lvert g^{(q=5)}_{\text{alt. pi}-\text{src, IRS}}\,(0,0)\right\rvert$
1	0.34131	0.44871	0.85003
2	0.25915	0.24786	0.42516
3	0.24603	0.19535	0.21226
4	0.24679	0.19819	0.63621

Table 6.7: Amplification factor averages and its absolute value for $(\Phi_x, \Phi_y) = (0,0)$ related to the number of multigrid levels considering a destructive source term coefficient ($S = -1000$) — multigrid analysis

However, both Fig. 6.45 and Tab. 6.7 show a degraded damping of very low–frequency modes for a 4 level V–cycle: While $\left\lvert g^{(q=5)}_{\text{pi}-\text{src, IRS}}\,(0,0)\right\rvert = 0.046422$ is the result for the standard linearization for this multigrid cycle, the corresponding value for the alternative linearization equals $\left\lvert g^{(q=5)}_{\text{alt. pi}-\text{src, IRS}}\,(0,0)\right\rvert = 0.63621$.
As this result indicates no improvement for the convergence of computational results based on the alternative linearization of the point implicitly treated source term is to be expected. The corresponding computational results are discussed in the following section.

6.2.3.3 Computational Results with Alternative Linearization

Despite the improved damping properties for low–frequency modes due to the alternative linearization on the one hand and the deteriorated damping properties for very low–frequency modes in case of a four level V–cycle on the other hand as presented in the previous section no significant changes in the convergence behaviour has been found.
Figs. 6.46 – 6.48 show the convergence obtained when combining all improvements obtained previously — singlegrid for the turbulence equations, a separate time step and a minimized coupling strategy — with the alternatively linearized source term. Nearly no differences are visible compared to Figs. 6.39 – 6.41. The largest difference found is that the bump in the convergence of the ω residual at $\mathrm{Re} = 60 \cdot 10^6$, Fig. 6.41, appears earlier with the alternative linearization than with the standard linearization, Fig. 6.41.
Other combinations — applying multigrid to the turbulence equations instead of singlegrid, using the same time step for RANS and turbulence equations or taking into account the standard coupling approach — showed less obvious differences in the convergence behaviour of all residuals.

6.2.3.4 Conclusions for Alternative Linearization

Although the Fourier analysis of the alternative linearization of the implicitly treated turbulence source term showed improved damping properties for low–frequency modes the computational results did not reproduce these improvements. Two reasons are possible for this behaviour. The one is that the damping properties for very low–frequency modes degraded when applying four multigrid levels within the Fourier analysis. The other reason is that the alternative linearization only influences the damping properties in case of destructive source terms. Following this idea it is natural that the convergence behaviour of the investigated test cases is mainly influenced by productive source terms.

6.2.4 Conclusions from Improvements to the Basic Scheme

Convergence to machine accuracy has been obtained for the density residual only if the coupling between RANS and turbulence equations has been minimized. That is the turbulent kinetic energy k is not taken into account within the RANS equations. Thus the turbulence equations influence the RANS equations only via the eddy viscosity.
The convergence of the turbulence residuals has been improved by modifying i. e. enlarging the time step for the turbulence equations. This improvement has been brought to bear only in combination with a singlegrid treatment for the turbulence equations as concluded in Sect. 6.1. However, at $\mathrm{Re} = 100 \cdot 10^6$ further robustness improvements are still necessary.
Nearly no changes in convergence have been found for the linearization of the implicitly treated source term before the Runge–Kutta method is applied.
The first two choices are taken as new defaults for the basic scheme on which an implicit treatment of the turbulence equations is introduced in the following section. The last one has been skipped.

Some probably successful approaches have not been implemented like the Runge–Kutta subiteration technique for the turbulence equations as proposed by Venkatakrishnan [127] and Couaillier et al.[27]. This technique especially has not been implemented since it is based on a singlegrid treatment of the turbulence equations as well. The implicit approaches to integrate the turbulence equations in time presented in the following sections have been found to be computationally as efficient as one Runge–Kutta step. Thus performing more Runge–Kutta steps on the finest grid for the turbulence equations would degrade the computational efficiency of the overall method.

6.3 Implicit Treatment of Turbulence Equations

An implicit time stepping scheme for the volume averaged scalar model equation, Eq. (3.1), is introduced below. Overviews of implicit approaches to solve the Navier–Stokes equations are given e. g. by Briley and McDonald [15] or by Venkatakrishnan [126]. For the implicit treatment of the turbulence equations the approximate factorization approach has been chosen.

The following section gives a detailed derivation of a general implicit scheme. Three possible approximations to the matrix of the linearized implicit scheme derived in Sect. 6.3.1 are given afterwards: A point implicit scheme according to Mavriplis [80] is presented in Sect. 6.3.2. A j–line implicit scheme is introduced in Sect. 6.3.3 while in Sect. 6.3.4 a diagonal dominant ADI (DDADI) scheme is defined. Each of these three sections comprises a first subsection for the derivation of the corresponding scheme together with its influences on the time stepping scheme in use. The Fourier analysis of the resulting time stepping scheme is discussed in a second subsection whereas a third subsection presents some computational results based on the conclusions drawn in Sect. 6.1 and Sect. 6.2. A fourth subsection gives first conclusions.
The use of implicit formulations of boundary conditions is discussed in Sect. 6.3.5. Concluding remarks on the implicit schemes are given in Sect. 6.3.6.
The extension of the implicit schemes to unsteady computations on the basis of the dual time stepping introduced by Jameson [52] is presented in Sect. A.4.

6.3.1 General Implicit Scheme

In Sect. 3.2.1 an explicit time discretization for the scalar model equation has been introduced based on the operator notation for the spatial discretization derived in Sect. 3.1:

$$\frac{W_{i,j}^{n+1} - W_{i,j}^{n}}{\Delta t_{i,j}} = \frac{1}{\widetilde{\mathcal{V}}_{i,j}} \left(Z_c \circ + Z_v \circ + Z_S \circ \right) W_{i,j}^{n} \,. \tag{3.52}$$

A point implicit treatment of the source term has been introduced in Sect. 3.2.2.2 as

$$\frac{W_{i,j}^{n+1} - W_{i,j}^{n}}{\tau_{i,j}} = \left(Z_c \circ + Z_v \circ + Z_S \circ \mathcal{T} \circ \right) W_{i,j}^{n} \tag{3.63}$$

with the time step per volume τ defined as in Eq. (3.76). From this basis it is easy to

denote a more implicit scheme by applying the time shift operator $\mathcal{T}\circ$ to the convective and the viscous terms also:

$$\begin{aligned}\frac{W_{i,j}^{n+1}-W_{i,j}^{n}}{\tau_{i,j}} &= (Z_c\circ\,\mathcal{T}\circ\,+Z_v\circ\,\mathcal{T}\circ\,+Z_S\circ\,\mathcal{T}\circ\,)\,W_{i,j}^{n}\\ &= (Z_c\circ\,+Z_v\circ\,+Z_S\circ\,)\,\mathcal{T}\circ\,W_{i,j}^{n}\,. \end{aligned}\tag{6.17}$$

This equation is — with respect to the definition of $\mathcal{T}\circ$, Eq. (3.64) — equivalent to

$$\frac{W_{i,j}^{n+1}-W_{i,j}^{n}}{\tau_{i,j}} = (Z_c\circ\,+Z_v\circ\,+Z_S\circ\,)\,W_{i,j}^{n+1}\tag{6.18}$$

which shows more directly the implicitness of the formulation. Defining

$$\Delta W_{i,j} = W_{i,j}^{n+1}-W_{i,j}^{n}\tag{6.19}$$

and linearizing Eq. (6.17) or Eq. (6.18) around $W_{i,j}^{n}$ leads to

$$\frac{\Delta W_{i,j}}{\tau_{i,j}} = Z\circ\,W_{i,j}^{n}+\left[Z_c\circ\,+Z_v\circ\,+\frac{\partial\left(Z_S\circ\,W_{i,j}^{n}\right)}{\partial W}\right]\Delta W_{i,j}\tag{6.20}$$

since $Z_c\circ$ and $Z_v\circ$ as presented in Sect. 3.1.1 resp. Sect. 3.1.2 are linear operators. $Z\circ$ is defined as in Sect. 3.3.3.1 to represent the sum of the spatial discretization operators:

$$Z\circ\,=Z_c\circ\,+Z_v\circ\,+Z_S\circ\,.\tag{3.86}$$

Resolving Eq. (6.20) for $\Delta W_{i,j}$ results in

$$\left(\frac{1}{\tau_{i,j}}-Z_c\circ\,-Z_v\circ\,-\frac{\partial\left(Z_S\circ\,W_{i,j}^{n}\right)}{\partial W}\right)\Delta W_{i,j} = Z\circ\,W_{i,j}^{n}\,.\tag{6.21}$$

From Sect. 3.1 the spatial discretizations, i. e. $Z\circ$ together with $Z_c\circ$ and $Z_v\circ$, are known and in Sect. 3.2.2 the linearization of the implicitly treated source term, i. e. the derivation of $\partial\left(Z_S\circ\,W_{i,j}^{n}\right)/\partial W$, has been discussed.

Considering in Eq. (6.21) not only one point (i,j) but all points results in a matrix vector multiplication

$$\mathsf{M}\overrightarrow{\Delta W} = Z\circ\,\overrightarrow{W^{n}}\tag{6.22}$$

with

$$\overrightarrow{W^{n}} = \begin{pmatrix}\vdots\\ W_{i-1,j}^{n}\\ W_{i,j}^{n}\\ W_{i+1,j}^{n}\\ \vdots\end{pmatrix}.\tag{6.23}$$

$\overrightarrow{\Delta W}$ is defined analogously.

While the reciprocal of the time step per volume, τ, and the approximated source term Jacobian contribute to the main diagonal of the matrix M only, the operators of the spatial discretization also fill the off diagonal entries of this matrix. This can be seen from a matrix notation of these operators presented in the following subsections.

6.3.1.1 Matrix Notation of the Spatial Discretization Operator of the Inviscid Terms

The operator $Z_c\circ$ has been defined in Sect. 3.1.1 as the sum of its one dimensional counterparts:

$$Z_c\circ \;=\; Z_c|^i\circ \;+\; Z_c|^j\circ \;. \tag{3.28}$$

A matrix notation is devised in the following for both one dimensional operator parts before the matrix of the operator $Z_c\circ$ is derived.

6.3.1.1.1 Matrix Notation for $\boldsymbol{Z_c\big|^i\circ}$

The discretization operator of the inviscid terms in i–direction has been given as

$$\begin{aligned} Z_c|^i\circ \;\;=\;\; & \frac{1}{2}\mathcal{C}_{i-\frac{1}{2},j}\left[q_{\vec{n};\,i-\frac{1}{2},j}\left(\mathcal{I}\circ\,+\mathcal{E}_i^{-1}\circ\right)-|q|_{\vec{n};\,i-\frac{1}{2},j}\left(\mathcal{I}\circ\,-\mathcal{E}_i^{-1}\circ\right)\right] \\ & -\frac{1}{2}\mathcal{C}_{i+\frac{1}{2},j}\left[q_{\vec{n};\,i+\frac{1}{2},j}\left(\mathcal{E}_i\circ\,+\mathcal{I}\circ\right)-|q|_{\vec{n};\,i+\frac{1}{2},j}\left(\mathcal{E}_i\circ\,-\mathcal{I}\circ\right)\right]\;. \end{aligned} \tag{3.26}$$

This operator can be simplified by collecting the coefficients

$$Z_c|^i\circ \;=\; Z_c|^i\Big[^{\mathcal{L}}_{i,j}\cdot\mathcal{E}_i^{-1}\circ \;+\; Z_c|^i\Big[^{\mathcal{D}}_{i,j}\cdot\mathcal{I}\circ \;+\; Z_c|^i\Big[^{\mathcal{U}}_{i,j}\cdot\mathcal{E}_i\circ\;. \tag{6.24}$$

This notation shows that the application of $Z_c|^i\circ$ to $W_{i,j}$ can be written as a product of a row vector containing the discretization coefficients $Z_c|^i\Big[^{*}_{i,j}$ on the one hand and a column vector of point data contributing to the discretization on the other:

$$Z_c|^i\circ W_{i,j} = \left(\; Z_c|^i\Big[^{\mathcal{L}}_{i,j}\;,\quad Z_c|^i\Big[^{\mathcal{D}}_{i,j}\;,\quad Z_c|^i\Big[^{\mathcal{U}}_{i,j}\right)\bullet\begin{pmatrix} W_{i-1,j}\\ W_{i,j}\\ W_{i+1,j}\end{pmatrix}. \tag{6.25}$$

The discretization coefficients $Z_c|^i\Big[^{*}_{i,j}$ can be easily derived from Eq. (3.26) as

$$Z_c|^i\Big[^{\mathcal{L}}_{i,j} \;=\; \frac{1}{2}\,\mathcal{C}_{i-\frac{1}{2},j}\left[q_{\vec{n};\,i-\frac{1}{2},j}+|q|_{\vec{n};\,i-\frac{1}{2},j}\right], \tag{6.26a}$$

$$Z_c|^i\Big[^{\mathcal{D}}_{i,j} \;=\; \frac{1}{2}\,\mathcal{C}_{i-\frac{1}{2},j}\left[q_{\vec{n};\,i-\frac{1}{2},j}-|q|_{\vec{n};\,i-\frac{1}{2},j}\right]-\frac{1}{2}\,\mathcal{C}_{i+\frac{1}{2},j}\left[q_{\vec{n};\,i+\frac{1}{2},j}+|q|_{\vec{n};\,i+\frac{1}{2},j}\right], \tag{6.26b}$$

$$Z_c|^i\Big[^{\mathcal{U}}_{i,j} \;=\; -\frac{1}{2}\,\mathcal{C}_{i+\frac{1}{2},j}\left[q_{\vec{n};\,i+\frac{1}{2},j}-|q|_{\vec{n};\,i+\frac{1}{2},j}\right]. \tag{6.26c}$$

Applying $Z_c|^i\circ$ to all points instead of a single point results in a matrix vector multiplication. The corresponding matrix M_c^i can be constructed as the sum of three submatrices built from the discretization coefficients given above:

$$\mathsf{M}_c^i = \mathsf{L}_c^i + \mathsf{D}_c^i + \mathsf{U}_c^i\,. \tag{6.27}$$

with a lower triangular matrix L_c^i, a diagonal matrix D_c^i and an upper triangular matrix U_c^i. The main diagonal of M_c^i is represented by

$$\mathsf{D}_c^i = \begin{pmatrix} \ddots & & \\ & Z_c|^i\big[_{i,j}^{\mathcal{D}} & \\ & & \ddots \end{pmatrix} . \tag{6.28}$$

The lower and upper triangular matrices, L_c^i and U_c^i, are built of $Z_c|^i\big[_{i,j}^{\mathcal{L}}$ and $Z_c|^i\big[_{i,j}^{\mathcal{U}}$ respectively.
As mentioned in Sect. 3.1 the FLOWer code uses two ghost cells at block boundaries. Thus

$$Z_c|^i\big[_{i_{\min},j}^{\mathcal{L}} \;=\; 0 \;=\; Z_c|^i\big[_{i_{\max},j}^{\mathcal{U}} \qquad \forall\, j = j_{\min}, \ldots, j_{\max}\,. \tag{6.29}$$

are the only boundary conditions to be applied to the coefficients of the discretization operator for the inviscid terms in i–direction with $i_{\min}$ and $i_{\max}$ being the indices of the outermost ghost points at the lower resp. upper end of the i–direction. Since the points $(i_{\min}-1, j)$ and $(i_{\max}+1, j)$ do not exist they must not contribute to the discretization at their neighboring points. This is assured by Eq. (6.29).
From Eq. (6.29) follows that L_c^i and U_c^i consist of $(j_{\max} - j_{\min})$ blocks. These blocks are defined as follows:

$$\mathcal{L}_c^i\big|_j = \begin{pmatrix} 0 & 0 & & \\ Z_c|^i\big[_{i_{\min}+1,j}^{\mathcal{L}} & \ddots & \ddots & \\ & \ddots & \ddots & 0 \\ & & Z_c|^i\big[_{i_{\max},j}^{\mathcal{L}} & 0 \end{pmatrix} \tag{6.30}$$

and

$$\mathcal{U}_c^i\big|_j = \begin{pmatrix} 0 & Z_c|^i\big[_{i_{\min},j}^{\mathcal{U}} & & \\ 0 & \ddots & \ddots & \\ & \ddots & \ddots & Z_c|^i\big[_{i_{\max}-1,j}^{\mathcal{U}} \\ & & 0 & 0 \end{pmatrix} \tag{6.31}$$

for all $j = j_{\min}, \ldots, j_{\max}$. These submatrices are used to define the remaining summands

$$\mathsf{L}_c^i = \begin{pmatrix} \ddots & & \\ & \mathcal{L}_c^i|_j & \\ & & \ddots \end{pmatrix} \tag{6.32}$$

and

$$\mathsf{U}_c^i = \begin{pmatrix} \ddots & & \\ & \mathcal{U}_c^i|_j & \\ & & \ddots \end{pmatrix} \tag{6.33}$$

for Eq. (6.27). It can easily be seen that M_c^i is a tridiagonal matrix.

6.3.1.1.2 Matrix Notation for $Z_c|^j \circ$

A matrix M_c^j analogous to Eq. (6.27) can be derived for the discretization operator of the j–direction of the convection terms, $Z_c|^j \circ$. This operator, Eq. (3.27), can be simplified by collecting the coefficients, too,

$$Z_c|^j \circ \;=\; Z_c|^j \Big[^{\mathcal{L}}_{i,j} \cdot \mathcal{E}_j^{-1} \circ \;+\; Z_c|^j \Big[^{\mathcal{D}}_{i,j} \cdot \mathcal{I} \circ \;+\; Z_c|^j \Big[^{\mathcal{U}}_{i,j} \cdot \mathcal{E}_j \circ \;. \tag{6.34}$$

The coefficients defining the operator $Z_c|^j \circ$ in this way are given by

$$Z_c|^j \Big[^{\mathcal{L}}_{i,j} \;=\; \frac{1}{2}\,\mathcal{C}_{i,j-\frac{1}{2}} \left[q_{\vec{n};\,i,j-\frac{1}{2}} + |q|_{\vec{n};\,i,j-\frac{1}{2}} \right] , \tag{6.35a}$$

$$Z_c|^j \Big[^{\mathcal{D}}_{i,j} \;=\; \frac{1}{2}\,\mathcal{C}_{i,j-\frac{1}{2}} \left[q_{\vec{n};\,i,j-\frac{1}{2}} - |q|_{\vec{n};\,i,j-\frac{1}{2}} \right] - \frac{1}{2}\,\mathcal{C}_{i,j+\frac{1}{2}} \left[q_{\vec{n};\,i,j+\frac{1}{2}} + |q|_{\vec{n};\,i,j+\frac{1}{2}} \right] , \tag{6.35b}$$

$$Z_c|^j \Big[^{\mathcal{U}}_{i,j} \;=\; -\frac{1}{2}\,\mathcal{C}_{i,j+\frac{1}{2}} \left[q_{\vec{n};\,i,j+\frac{1}{2}} - |q|_{\vec{n};\,i,j+\frac{1}{2}} \right] . \tag{6.35c}$$

Again some of these discretization coefficients become zero at $j_{\min}$ resp. at $j_{\max}$:

$$Z_c|^j \Big[^{\mathcal{L}}_{i,j_{\min}} \;=\; 0 \;=\; Z_c|^j \Big[^{\mathcal{U}}_{i,j_{\max}} \qquad \forall\, i = i_{\min}, \ldots, i_{\max} \,. \tag{6.36}$$

Splitting M_c^j in the same way as M_c^i, i. e.

$$\mathsf{M}_c^j = \mathsf{L}_c^j + \mathsf{D}_c^j + \mathsf{U}_c^j \tag{6.37}$$

gives rise to analogous definitions for the lower triangular matrix L_c^j, the diagonal matrix D_c^j and the upper triangular matrix U_c^j.
D_c^j is given by

$$\mathsf{D}_c^j = \begin{pmatrix} \ddots & & \\ & Z_c|^j \Big[^{\mathcal{D}}_{i,j} & \\ & & \ddots \end{pmatrix} . \tag{6.38}$$

The blocks $\mathcal{L}_c^j|_i$, $i = i_{\min}, \ldots, i_{\max}$, building the lower triangular matrix L_c^j in an equivalent way to Eq. (6.32) are given as

$$\mathcal{L}_c^j\Big|_i = \begin{pmatrix} 0 & \cdots & 0 & 0 & & \\ \vdots & \ddots & & \ddots & \ddots & \\ 0 & & \ddots & & \ddots & 0 \\ Z_c|^j\left[{}^{\mathcal{L}}_{i,j_{\min}+1}\right. & \ddots & & \ddots & & 0 \\ & \ddots & \ddots & & \ddots & \vdots \\ & & Z_c|^j\left[{}^{\mathcal{L}}_{i,j_{\max}}\right. & 0 & \cdots & 0 \end{pmatrix}. \tag{6.39}$$

This larger bandwidth compared to $\mathcal{L}_c^i|_j$, Eq. (6.30), can be easily derived from the application of the spatial discretization operator in j–direction $Z_c|^j\circ$ to $W_{i,j}$

$$Z_c|^j \circ W_{i,j} = \left(Z_c|^j\left[{}^{\mathcal{L}}_{i,j}\right. , \quad Z_c|^j\left[{}^{\mathcal{D}}_{i,j}\right. , \quad Z_c|^j\left[{}^{\mathcal{U}}_{i,j}\right. \right) \bullet \begin{pmatrix} W_{i,j-1} \\ W_{i,j} \\ W_{i,j+1} \end{pmatrix}. \tag{6.40}$$

The vector of W's is still the same (cmp. Eq. (6.23)) but when applying $Z_c|^j\circ$ to $\overrightarrow{W}$ different components are taken into account compared to the application of $Z_c|^i\circ$:

$$\overrightarrow{W} = \begin{pmatrix} \vdots \\ W_{i,j-1} \\ \vdots \\ W_{i,j} \\ \vdots \\ W_{i,j+1} \\ \vdots \end{pmatrix}. \tag{6.41}$$

The analogous blocks building the upper triangular matrix U_c^j, i. e. $\mathcal{U}_c^j\big|_i$ for $i = i_{\min}, \ldots, i_{\max}$, read

$$\mathcal{U}_c^j\Big|_i = \begin{pmatrix} 0 & \cdots & 0 & Z_c|^j\lceil^{\mathcal{U}}_{i,j_{\min}} & & \\ \vdots & \ddots & & \ddots & \ddots & \\ 0 & & \ddots & & \ddots & Z_c|^j\lceil^{\mathcal{U}}_{i,j_{\max}-1} \\ 0 & \ddots & & \ddots & & 0 \\ & \ddots & \ddots & & \ddots & \vdots \\ & & 0 & 0 & \cdots & 0 \end{pmatrix}. \tag{6.42}$$

Insertion of $\mathcal{U}_c^j\big|_i$ to U_c^j and $\mathcal{L}_c^j\big|_i$ to L_c^j in analogy to Eqs. (6.32) and (6.33) and applying Eq. (6.37) shows that M_c^j is a tridiagonal matrix, too, but the bandwidth is larger than for M_c^i: The bandwidth of M_c^j equals $i_{max} + 1$.

6.3.1.1.3 Matrix Notation for $Z_c\circ$

The matrix representing the full operator $Z_c\circ$, M_c, equals the sum of the matrices M_c^* derived in the previous sections:

$$\mathsf{M}_c = \mathsf{M}_c^i + \mathsf{M}_c^j. \tag{6.43}$$

A different notation related to the derivation of the matrices M_c^* is

$$\mathsf{M}_c = \mathsf{L}_c + \mathsf{D}_c + \mathsf{U}_c \tag{6.44}$$

with a lower triangular matrix L_c, a diagonal matrix D_c and an upper triangular matrix U_c. These matrices are defined as the sums of the corresponding parts of the matrices M_c^*, i. e.

$$\begin{aligned} \mathsf{L}_c &= \mathsf{L}_c^i + \mathsf{L}_c^j, \\ \mathsf{D}_c &= \mathsf{D}_c^i + \mathsf{D}_c^j \quad \text{and} \\ \mathsf{U}_c &= \mathsf{U}_c^i + \mathsf{U}_c^j. \end{aligned} \tag{6.45}$$

This notation directly illustrates that the matrix M_c is pentadiagonal although built from tridiagonal matrices. The reason is given by the different bandwidths of these tridiagonal matrices.

6.3.1.2 Matrix Notation of the Spatial Discretization Operator of the Viscous Terms

In Sect. 3.1.2 the operator $Z_v\circ$ for the spatial discretization of the viscous terms has been introduced in terms of a thin layer approximation:

(3.42) $$Z_v\circ \;=\; Z_v|^j\circ \;.$$

This operator reads

(3.41)
$$\begin{aligned} Z_v|^j\circ \;\; &= \;\; \mu_{i,j+\frac{1}{2}}\,(\mathcal{C}\vec{n})^*_{i,j+\frac{1}{2}}\bullet\vec{n}_{i,j+\frac{1}{2}}\,\mathcal{C}_{i,j+\frac{1}{2}}\left(\mathcal{E}_j\circ\;-\mathcal{I}\circ\right)\\ &\quad -\mu_{i,j-\frac{1}{2}}\,(\mathcal{C}\vec{n})^*_{i,j-\frac{1}{2}}\bullet\vec{n}_{i,j-\frac{1}{2}}\,\mathcal{C}_{i,j-\frac{1}{2}}\left(\mathcal{I}\circ\;-\mathcal{E}_j^{-1}\circ\right)\end{aligned}$$

with $(\mathcal{C}\vec{n})^*_{i,j\pm\frac{1}{2}}$ defined as in Eq. (3.36).
In analogy to the derivation of the matrix notation for the discretization of the convective terms the operator $Z_v|^j\circ$ is rewritten as

$$Z_v|^j\circ \;=\; Z_v|^j\Big[^{\mathcal{L}}_{i,j}\cdot\mathcal{E}_j^{-1}\circ \;+\; Z_v|^j\Big[^{\mathcal{D}}_{i,j}\cdot\mathcal{I}\circ \;+\; Z_v|^j\Big[^{\mathcal{U}}_{i,j}\cdot\mathcal{E}_j\circ \tag{6.46}$$

with the coefficients defining the operator $Z_v|^j\circ$ given by

$$Z_v|^j\Big[^{\mathcal{L}}_{i,j} \;=\; \mu_{i,j-\frac{1}{2}}\,(\mathcal{C}\vec{n})^*_{i,j-\frac{1}{2}}\bullet\vec{n}_{i,j-\frac{1}{2}}\,\mathcal{C}_{i,j-\frac{1}{2}}\,, \tag{6.47a}$$

$$Z_v|^j\Big[^{\mathcal{D}}_{i,j} \;=\; -\mu_{i,j-\frac{1}{2}}\,(\mathcal{C}\vec{n})^*_{i,j-\frac{1}{2}}\bullet\vec{n}_{i,j-\frac{1}{2}}\,\mathcal{C}_{i,j-\frac{1}{2}} - \mu_{i,j+\frac{1}{2}}\,(\mathcal{C}\vec{n})^*_{i,j+\frac{1}{2}}\bullet\vec{n}_{i,j+\frac{1}{2}}\,\mathcal{C}_{i,j+\frac{1}{2}}\,, \tag{6.47b}$$

$$Z_v|^j\Big[^{\mathcal{U}}_{i,j} \;=\; \mu_{i,j+\frac{1}{2}}\,(\mathcal{C}\vec{n})^*_{i,j+\frac{1}{2}}\bullet\vec{n}_{i,j+\frac{1}{2}}\,\mathcal{C}_{i,j+\frac{1}{2}}\,. \tag{6.47c}$$

Once more some of these discretization coefficients become zero at $j_{\min}$ resp. at $j_{\max}$:

$$Z_v|^j\Big[^{\mathcal{L}}_{i,j_{\min}} \;=\; 0 \;=\; Z_v|^j\Big[^{\mathcal{U}}_{i,j_{\max}} \qquad \forall\, i=i_{\min},\ldots,i_{\max}\,. \tag{6.48}$$

Thus M_v^j has a block structure containing tridiagonal blocks, too. The bandwidth is the same as for M_c^j since application of $Z_v|^j\circ$ to $W_{i,j}$ takes contributions from the same points as the application of $Z_c|^j\circ$ does. For this compare Eq. (6.40) to

$$Z_v|^j\circ W_{i,j} = \left(\; Z_v|^j\Big[^{\mathcal{L}}_{i,j}\;,\quad Z_v|^j\Big[^{\mathcal{D}}_{i,j}\;,\quad Z_v|^j\Big[^{\mathcal{U}}_{i,j}\right)\bullet\begin{pmatrix} W_{i,j-1}\\ W_{i,j}\\ W_{i,j+1}\end{pmatrix}. \tag{6.49}$$

For any i the tridiagonal blocks building M_v^j are the same as for M_c^j except that now the matrix entries are based on the discretization of the viscous terms in j–direction:

$$\mathcal{M}_v^j\big|_i = \mathcal{L}_v^j\big|_i + \mathcal{D}_v^j\big|_i + \mathcal{U}_v^j\big|_i\,. \tag{6.50}$$

Using this abbreviation M_v^j resp. M_v is given as

$$\mathsf{M}_v = \mathsf{M}_v^j = \begin{pmatrix} \ddots & & \\ & \mathcal{M}_v^j\big|_i & \\ & & \ddots\end{pmatrix}. \tag{6.51}$$

The diagonal matrices $\mathcal{D}_v^j\big|_i$ in Eq. (6.50) are given as

$$\mathcal{D}_v^j\Big|_i = \begin{pmatrix} {Z_v}^j\Big[^{\mathcal{D}}_{i,j_{\min}} & & \\ & \ddots & \\ & & {Z_v}^j\Big[^{\mathcal{D}}_{i,j_{\max}} \end{pmatrix} \tag{6.52}$$

for $i = i_{\min}, \ldots, i_{\max}$. The lower triangular blocks read for the same range of i

$$\mathcal{L}_v^j\Big|_i = \begin{pmatrix} 0 & \cdots & 0 & 0 & & \\ \vdots & \ddots & & \ddots & \ddots & \\ 0 & & \ddots & & \ddots & 0 \\ {Z_v}^j\Big[^{\mathcal{L}}_{i,j_{\min}+1} & \ddots & & \ddots & & 0 \\ & \ddots & \ddots & & \ddots & \vdots \\ & & {Z_v}^j\Big[^{\mathcal{L}}_{i,j_{\max}} & 0 & \cdots & 0 \end{pmatrix}. \tag{6.53}$$

Also the upper triangular blocks have the same appearance as their analogue for the convection in j–direction:

$$\mathcal{U}_v^j\Big|_i = \begin{pmatrix} 0 & \cdots & 0 & {Z_v}^j\Big[^{\mathcal{U}}_{i,j_{\min}} & & \\ \vdots & \ddots & & \ddots & \ddots & \\ 0 & & \ddots & & \ddots & {Z_v}^j\Big[^{\mathcal{U}}_{i,j_{\max}-1} \\ 0 & \ddots & & \ddots & & 0 \\ & \ddots & \ddots & & \ddots & \vdots \\ & & 0 & 0 & \cdots & 0 \end{pmatrix}. \tag{6.54}$$

6.3.1.3 Matrix Notation of the Spatial Discretization Operator of the Source Term

As noted in Sect. 3.1.3 the source terms are not spatially discretized. Their spatial discretization operator has been given as

(3.46) $$Z_S \circ \; = \widetilde{\mathcal{V}}\Big|_{*,*} S\left(\mathcal{I} \circ\right)$$

with S being a nonlinear function. In Sect. 3.2.2 the (point) implicit treatment of the

source terms has been introduced. Within this context an approximated source term Jacobian has been derived as the only contribution to the left hand side.
Deriving a matrix notation for this contribution to the matrix M in Eq. (6.22) along with the matrices M_* derived in the previous sections results in

$$\mathsf{M}_S = \mathsf{L}_S + \mathsf{D}_S + \mathsf{U}_S \tag{6.55}$$

with the lower resp. upper triangular matrix being null matrices:

$$\mathsf{L}_S = \mathsf{U}_S = \begin{pmatrix} \ddots & & \\ & 0 & \\ & & \ddots \end{pmatrix} . \tag{6.56}$$

Therefore they are no longer considered. Thus any contribution of the source terms to M may be only to the main diagonal, i. e.

$$\mathsf{M}_S = \mathsf{D}_S = \begin{pmatrix} \ddots & & \\ & \dfrac{\partial (Z_S \circ W^n)}{\partial W} & \\ & & \ddots \end{pmatrix} . \tag{6.57}$$

6.3.1.4 Matrix Notation of the Spatial Discretization Operator of the Model Equation including a Source Term

In this section the matrix of the spatial discretization operator $Z\circ$ for the complete model equation including the source term is derived. In analogy to the definition of $Z\circ$, Eq. (3.86), the corresponding matrix is the sum of the matrices related to the discretization operators $Z_c\circ$, $Z_v\circ$ and $Z_S\circ$:

$$\mathsf{M} = \frac{1}{\tau} \cdot \mathsf{I} - \mathsf{M}_c - \mathsf{M}_v - \mathsf{M}_S \, . \tag{6.58}$$

I denotes the identity matrix. Like the matrices for the discretization operators $Z_*\circ$ the matrix M can also be written as a sum

$$\mathsf{M} = \mathsf{L} + \mathsf{D} + \mathsf{U} \tag{6.59}$$

of a lower and upper triangular matrix, respectively, and a diagonal matrix. These matrices are built by collecting the corresponding matrices derived in the preceding sections.
The lower triangular matrix is given as

$$\mathsf{L} = -\mathsf{L}_c - \mathsf{L}_v \tag{6.60}$$

while its upper triangular counterpart reads

$$\mathsf{U} = -\mathsf{U}_c - \mathsf{U}_v \, . \tag{6.61}$$

The diagonal matrix D on the other hand does not only consist of the contributions of the discretization operators but contains the reciprocal of the time step per volume τ — as defined in Eq. (3.76) — as well:

$$\mathsf{D} = \frac{1}{\tau} \cdot \mathsf{I} - \mathsf{D}_c - \mathsf{D}_v - \mathsf{D}_S \,. \tag{6.62}$$

In the end this equation completes the definition of the matrix M as used in Eq. (6.22).

In order to be able to adjust the time step per volume τ to the stability range of the implicit time stepping schemes for the turbulence equations discussed in the following the input parameter `CFLTU` has been introduced to the FLOWer code. `CFLTU` denotes the CFL number used for the time integration of the turbulence equations. Its default setting is `CFLTU` = `CFL`.

6.3.2 Point Implicit Treatment of Whole Equation

Mavriplis and Martinelli [80] suggested to treat not only the source term point implicitly as implemented in FLOWer but also convection and diffusion terms. By the implicit treatment of diffusion and convection terms of turbulence the diagonal dominance of the resulting Jacobian matrix is said to be increased further. This approach is discussed in this section and its subsections, respectively.

6.3.2.1 Derivation of Point Implicit Treatment of Whole Equation

The point implicit scheme according to Mavriplis and Martinelli [80] differs from the standard point implicit scheme in FLOWer. FLOWer uses

$$\mathsf{M} \approx \frac{1}{\tau} \cdot \mathsf{I} - \mathsf{D}_S \tag{6.63}$$

as an approximation to the matrix M of the implicit scheme. The approach of Mavriplis and Martinelli takes also into account the diagonal matrices derived from the spatial discretization of convective and diffusive terms:

$$\mathsf{M} \approx \mathsf{D} = \frac{1}{\tau} \cdot \mathsf{I} - \mathsf{D}_c - \mathsf{D}_v - \mathsf{D}_S \,. \tag{6.64}$$

Both approaches are based on a multi stage time stepping scheme but they affect the Runge–Kutta scheme differently: The effect of the standard point implicit treatment in FLOWer on the Runge–Kutta scheme is that the stage coefficient α_q is modified as

(3.67)
$$\alpha_q^{\mathrm{pi-src}} = \frac{\alpha_q}{1 - \alpha_q \tau \dfrac{\partial\left(Z_S \circ W^{(0)}\right)}{\partial W}} \,. \tag{6.65}$$

The approach of Mavriplis et al. can be written in terms of a modification of the stage coefficient α_q, too. This modification reads

$$\alpha_q^{\mathrm{pi-eq}} = \frac{\alpha_q}{1 - \alpha_q \tau \mathsf{D}_{i,j}} \tag{6.66}$$

with the implicitly treated part of the (i,j)th entry in the matrix D, Eq. (6.62),

$$\mathsf{D}_{i,j} = Z_c|^i\Big[^{\mathcal{D}}_{i,j} + Z_c|^j\Big[^{\mathcal{D}}_{i,j} + Z_v|^j\Big[^{\mathcal{D}}_{i,j} + \left.\frac{\partial\left(Z_S \circ W^{(0)}\right)}{\partial W}\right|_{i,j} . \tag{6.66}$$

These different effects on the Runge–Kutta stage coefficient α_q are based on the linearization approach of the implicitly treated source term presented in Sect. 3.2.2.2.

Due to the assumption of given velocities the operator $Z_c\circ$ has been defined independently of the current Runge–Kutta stage. Hence $\alpha_q^{\mathrm{pi-eq}}$ represents an application of the Runge–Kutta scheme to the turbulence equations separated from its application to the RANS equations. This is in difference to the implementation in the FLOWer code. There RANS and turbulence equations are treated simultaneously on each Runge–Kutta stage.

The influence of the difference between the modified stage coefficients on the damping properties of the Runge–Kutta method is discussed in the following section.

6.3.2.2 Fourier Analysis of Point Implicit Treatment of Whole Equation

The amplification factor of the Runge–Kutta scheme with a point implicit treatment according to Mavriplis et al. is named $g^{(q=5)}_{\mathrm{pi-eq}}$. The subscript $_{\mathrm{pi-eq}}$ denotes the point implicit treatment of the whole model equation. $g^{(q=5)}_{\mathrm{pi-eq}}$ is derived in analogy to Sect. 4.3.2: This time $f^{(q=5)}_{\mathrm{pi-src}}$ is substituted in Eq. (4.38) by $f^{(q=5)}_{\mathrm{pi-eq}}$ yielding

$$g^{(q=5)}_{\mathrm{pi-eq}} = \frac{\widehat{W}^{n+1}}{\widehat{W}^{n}} = 1 + f^{(q=5)}_{\mathrm{pi-eq}}\,\tau\left(\mathcal{Z}_c + \mathcal{Z}_v + \mathcal{Z}_S\right) . \tag{6.67}$$

$f^{(q=5)}_{\mathrm{pi-eq}}$ is derived from $f^{(q=5)}_{\mathrm{pi-src}}$, Eq. (4.40), based on the modified stage coefficient $\alpha_q^{\mathrm{pi-eq}}$ as given in Eq. (6.65) and reads

$$\begin{aligned} f^{(q)}_{\mathrm{pi-eq}} \;=\; & \alpha_q \frac{1}{1-\alpha_q\tau\mathcal{D}_{i,j}} \\ & + \alpha_q\alpha_{q-1}\frac{1}{1-\alpha_q\tau\mathcal{D}_{i,j}}\,\frac{1}{1-\alpha_{q-1}\tau\mathcal{D}_{i,j}}\,\tau\mathcal{Z}_c \\ & + \ldots + \prod_{r=1}^{q}\alpha_r\frac{1}{1-\alpha_r\tau\mathcal{D}_{i,j}}\,(\tau\mathcal{Z}_c)^{q-1} . \end{aligned} \tag{6.68}$$

Here $\mathcal{D}_{i,j}$ denotes the Fourier symbol of $\mathsf{D}_{i,j}$, Eq. (6.66). Applying the assumptions made for Fourier analysis in Chap. 4 to the definition of $\mathsf{D}_{i,j}$ yields

$$\mathcal{D}_{i,j} = -u\Delta y - v\Delta x - 2\mu\frac{\Delta x}{\Delta y} + \min\left(0, \mathcal{Z}_S\right) . \tag{6.69}$$

Furthermore $g^{(q=5)}_{\mathrm{pi-eq}}$ can be easily extended by implicit residual smoothing in the same way used in Sect. 3.3.2 or Sect. 4.4.1 which leads to

$$g^{(q=5)}_{\mathrm{pi-eq,\,IRS}} = \frac{\widehat{W}^{n+1}}{\widehat{W}^{n}} = 1 + f^{(q=5)}_{\mathrm{pi-eq,\,IRS}}\,\tau\left(\mathcal{Z}_c + \mathcal{Z}_v + \mathcal{Z}_S\right) \tag{6.70}$$

with $f^{(q)}_{\mathrm{pi-eq,\,IRS}}$ given as

$$
\begin{aligned}
f^{(q)}_{\mathrm{pi-eq,\,IRS}} \;=\; \mathcal{B}\Bigg[& \frac{\alpha_q}{1-\alpha_q\tau\mathcal{D}_{i,j}} \\
& + \frac{\alpha_q}{1-\alpha_q\tau\mathcal{D}_{i,j}}\,\frac{\alpha_{q-1}}{1-\alpha_{q-1}\tau\mathcal{D}_{i,j}}\,\mathcal{B}\tau\mathcal{Z}_c \\
& + \ldots + \prod_{r=1}^{q}\frac{\alpha_r}{1-\alpha_r\tau\mathcal{D}_{i,j}}\,(\mathcal{B}\tau\mathcal{Z}_c)^{q-1}\Bigg] .
\end{aligned}
\tag{6.71}
$$

The Fourier footprint of the amplification factor $g^{(q=5)}_{\mathrm{pi-eq,\,IRS}}$ with the same settings as used for the Fourier footprint of $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$ in Fig. 5.51 is shown in Fig. 6.49. For both cases a productive source term with $S = 1000$ is taken.
Even at first glance the worsened damping properties in Fig. 6.49 can be caught. They occur if $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$ is simply replaced by $g^{(q=5)}_{\mathrm{pi-eq,\,IRS}}$. Further adaptation to the increased implicitness of the time stepping scheme has been made by reducing or leaving out implicit residual smoothing. This improves the damping properties as shown in Fig. 6.50, the Fourier footprint of $g^{(q=5)}_{\mathrm{pi-eq}}$. An important point is that the CFL number has not been reduced for this analysis. Kroll and Jain [63] point out that by application of implicit residual smoothing the maximum but stable CFL number may be increased by a factor of about 2. Vice versa the CFL number should be reduced accordingly when leaving out implicit residual smoothing. While for Fig. 6.50 $\mathtt{CFL} = 3.75$ has been used, Fig. 6.51 shows the corresponding Fourier footprint with an unreduced CFL number of 7.5. Slight improvements can be observed for very high–frequency Φ_y–modes for the larger CFL number.
Despite the improvement obtained without implicit residual smoothing compared to $g^{(q=5)}_{\mathrm{pi-eq,\,IRS}}$ no improvement could be achieved according to the overall smoothing properties compared to $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$. This is shown in Tab. 6.8.

point implicit approach	$\overline{\left\lvert g_*^{(q=5)}\right\rvert}_{\mathrm{all}}$	$\overline{\left\lvert g_*^{(q=5)}\right\rvert}_{\mathrm{low}}$	$\left\lvert g_*^{(q=5)}\,(0,0)\right\rvert$
$* = \mathrm{pi-src,\ IRS}$	0.40208	0.57503	1.15
$* = \mathrm{pi-eq,\ IRS}$	0.86598	0.91298	1.0273
$* = \mathrm{pi-eq,\ CFL} = 3.75$	0.46862	0.79589	1.0231
$* = \mathrm{pi-eq,\ CFL} = 7.5$	0.44299	0.76287	1.0273

Table 6.8: Amplification factor averages and its absolute value for $(\Phi_x, \Phi_y) = (0, 0)$ related to different point implicit approaches considering a productive source term coefficient ($S = 1000$) — singlegrid analysis

Secondly it can be observed in Tab. 6.8 that the amplification factor for $(\Phi_x, \Phi_y) = (0, 0)$ is reduced treating the whole equation point implicitly. Its further reduction due to a smaller CFL number is counteracted by increases in $\overline{\left\lvert g^{(q=5)}_{\mathrm{pi-eq}}\right\rvert}_{\mathrm{all}}$ and $\overline{\left\lvert g^{(q=5)}_{\mathrm{pi-eq}}\right\rvert}_{\mathrm{low}}$.

However, the overall damping properties of a multigrid cycle with this extended implicit operator still lack improvement except for a four level V–cycle. This lack especially holds for low–frequency modes for which multigrid is supposed to improve damping. Comparing Tab. 6.9 to Tab. 5.7 yields this result. The corresponding Fourier footprints have not been included as they do not give more insight.

# multigrid level	$\overline{\left\lvert g_{\mathrm{pi-eq}}^{(q=5)}\right\rvert}_{\mathrm{all}}$	$\overline{\left\lvert g_{\mathrm{pi-eq}}^{(q=5)}\right\rvert}_{\mathrm{low}}$	$\left\lvert g_{\mathrm{pi-eq}}^{(q=5)}\,(0,0)\right\rvert$
1	0.44299	0.76287	1.0273
2	0.37311	0.58475	1.1128
3	0.36592	0.56283	1.3646
4	0.36796	0.57207	2.1169

Table 6.9: Amplification factor averages and its absolute value for $(\Phi_x, \Phi_y) = (0, 0)$ related to the number of multigrid levels considering a productive source term coefficient ($S = 1000$) — multigrid analysis

Although the amplification of $(\Phi_x, \Phi_y) = (0, 0)$ for a four level V–cycle is reduced by a factor of about 10 from $g_{\mathrm{pi-src,\,IRS}}^{(q=5)}$ to $g_{\mathrm{pi-eq}}^{(q=5)}$ applied to the model equation with a productive source term — $S = 1000$ — within this analysis no increase in robustness has been obtained for multigrid computations using the point implicit approach of Mavriplis et al. [80] as discussed among other computational results in the following section.

6.3.2.3 Computational Results with Point Implicit Treatment of Whole Equation

With the point implicit treatment of the whole turbulence equations as presented above several computations have been performed for the comparison to the point implicit treatment of the source terms only. The turbulence equations are not subject to the multigrid method and the enlarged time step for the turbulence equations is used for all computations. From Figs. 6.52 – 6.54 no general difference in convergence behaviour for both the density residual and the turbulence residuals can be seen. Only one thing has to be pointed out: The point implicit treatment according to Mavriplis et al. does level out for all residuals in case of minimized coupling and kept range for smoothing coefficients, $\varepsilon_i, \varepsilon_j \in [0.2, 1.2]$. Convergence to machine accuracy is only obtained if the implicit residual smoothing is reduced, i. e. $\varepsilon_i, \varepsilon_j \in [0.2, 1.0]$.

Comparing Figs. 6.55 – 6.57 to Figs. 6.39 – 6.41 yields that the point implicit treatment of the whole turbulence equations does not increase robustness nor convergence speed in a measurable amount over the considered range of Reynolds numbers.

Additional computations did not show any increase of the CFL number to be possible although the Fourier analysis in the previous section suggested this. However, not all possible settings of FLOWer input parameters related to the implicit residual smoothing have been checked. Further no separated parameters for the implicit smoothing of the turbulence equations have been implemented. Thus any change in the smoothing parameters of the FLOWer code also affects the RANS equations.

The point implicit treatment of the complete turbulence equations raises the CPU time per iteration about 6% on NEC SX–5 vector computer while the memory requirement of the implementation to the FLOWer code is still the same as for the standard point implicit treatment. This increase in CPU time is due to the additional calculation of the discretization coefficients $Z_c|^i\left[^{\mathcal{D}}_{i,j}\right.$, $Z_c|^j\left[^{\mathcal{D}}_{i,j}\right.$ and $Z_v|^j\left[^{\mathcal{D}}_{i,j}\right.$.

6.3.2.4 Conclusions for Point Implicit Treatment of Whole Equation

The point implicit treatment of convection and diffusion has been a by–product of the implementation of more implicit schemes presented below. Thus the point implicit matrix is calculated only once per Runge–Kutta step, i. e. for the first stage. This is suitable for source and diffusion terms as they are also evaluated once per Runge–Kutta step (for the first stage). Convection is evaluated on every Runge–Kutta stage but its implicit treatment is not updated accordingly. However, this is still in correspondence with the Fourier analysis. In contrast to the Fourier analysis the implicit residual smoothing has not been reduced significantly for the turbulence equations since its coefficients are used for both the RANS and the turbulence equations.

It has to be concluded that the current implementation of the point implicit treatment of the turbulence equations according to Mavriplis and Martinelli [80] does not show any improvements related to robustness or convergence speed compared to a point implicit treatment of the source term only. Additionally 6% more CPU time has to be spent for this approach on a NEC SX–5 vector computer.

6.3.3 J–Line Implicit Scheme

A better approximation of the matrix M of the fully implicit scheme than the diagonal matrix D as presented above is a matrix M^* taking into account one coordinate direction e. g. the wall normal direction. Such line implicit schemes are well known and have been presented by Venkatakrishnan [126, 127], Turkel et al. [122] as well as by others as time stepping schemes for the RANS equations.
As a thin layer approximation of the viscous fluxes is employed throughout this work the line to be treated implicitly should contain these fluxes in order to increase implicitness of the method seriously. Thus a j–line implicit treatment of the turbulence equations is discussed. Its derivation is done in the same notation as used throughout this work. Based on this a Fourier analysis has been performed. The corresponding results are presented before computational results using this implicit method are shown. Concluding remarks bring the section to a close.

6.3.3.1 Derivation of J–Line Implicit Scheme

An implicit treatment of the j–direction results in an approximation of M by some matrix M^j. This matrix is built by summing up all contributions to M derived from discretizations in j–direction, i. e.

$$\mathsf{M} \approx \mathsf{M}^j = \frac{1}{\tau} \cdot \mathsf{I} - \mathsf{M}_c^j - \mathsf{M}_v^j - \mathsf{M}_S \,. \tag{6.72}$$

The matrices M_*^j have been derived in Sect. 6.3.1. Splitting M^j in lower, diagonal and upper part reads

$$\mathsf{M}^j = \mathsf{L}^j + \mathsf{D}^j + \mathsf{U}^j \tag{6.73}$$

with

$$\mathsf{L}^j = -\mathsf{L}_c^j - \mathsf{L}_v^j\,, \tag{6.74}$$

$$\mathsf{D}^j = \frac{1}{\tau}\cdot\mathsf{I} - \mathsf{D}_c^j - \mathsf{D}_v^j - \mathsf{D}_S \tag{6.75}$$

and

$$\mathsf{U}^j = -\mathsf{U}_c^j - \mathsf{U}_v^j\,. \tag{6.76}$$

This approximation of M is a tridiagonal matrix. The resulting scheme

$$\mathsf{M}^j \cdot \overrightarrow{\Delta W} \approx Z \circ \overrightarrow{W^n} \tag{6.77}$$

approximates Eq. (6.22) and is solved by means of the Thomas' algorithm for tridiagonal systems [1, 46].

Sterner [112] uses this line implicit approach in the framework of a semi–implicit Runge–Kutta method.* By this semi–implicit Runge–Kutta method he treats the streamwise direction explicitly while the wall normal direction is treated implicitly. Turkel et al. [122] apply the same approach as a so–called preconditioner in order to gain a better conditioned problem to be solved.
A first idea to follow Sterner [112] and Turkel et al. [122] in the incorporation of the j–line implicit approach — not for the RANS equations but for the turbulence equations — in the explicit Runge–Kutta method has been investigated without success. As shown in the next section but one the direct solution of this approximated implicit scheme yields satisfying convergence and robustness improvements. Before those computational results are discussed this implicit scheme is subject to a Fourier analysis.

6.3.3.2 Fourier Analysis of J–Line Implicit Scheme

For the Fourier analysis an operator notation of the time stepping scheme applicable to a single grid point is needed. The line implicit scheme in Eq. (6.77) can be rewritten for one point as

$$M^j \circ W_{i,j}^{n+1} \approx \left(M^j \circ + Z \circ\right) W_{i,j}^n \tag{6.78}$$

i. e.

$$W_{i,j}^{n+1} \approx \mathcal{G}_{j\text{–line}} \circ W_{i,j}^n \tag{6.79}$$

with the operator for the j–line implicit time stepping scheme

$$\mathcal{G}_{j\text{–line}} \circ = \mathcal{I} \circ + \left(M^j \circ\right)^{-1} Z \circ\,. \tag{6.80}$$

*Sterner [112] abbreviates his "**S**emi–**I**mplicit **R**unge–**K**utta" method by "SIRK". This is in difference to Hairer and Wanner [45] who use the abbreviation "SIRK" for "**S**ingly **I**mplicit **R**unge–**K**utta" methods. A singly implicit Runge–Kutta method is a Runge–Kutta method with a one–point spectrum.

The definition of $M^j\circ$ can be summarized from the previous sections:

$$\begin{aligned} M^j\circ \;=\; & \left(-\,Z_c|^j\Big[^{\mathcal{L}}_{i,j} \;-\; Z_v|^j\Big[^{\mathcal{L}}_{i,j}\right)\mathcal{E}_j^{-1}\circ \\ & +\left(\frac{1}{\tau}-\,Z_c|^j\Big[^{\mathcal{D}}_{i,j} \;-\; Z_v|^j\Big[^{\mathcal{D}}_{i,j} \;-\; Z_S|^j\Big[^{\mathcal{D}}_{i,j}\right)\mathcal{I}\circ \\ & +\left(-\,Z_c|^j\Big[^{\mathcal{U}}_{i,j} \;-\; Z_v|^j\Big[^{\mathcal{U}}_{i,j}\right)\mathcal{E}_j\circ\;. \end{aligned} \tag{6.81}$$

Thus $M^j\circ$ has the same structure as the operator $B_j\circ$ given in Eq. (3.80). Before deriving the Fourier symbol $\mathcal{M}^j$ of the inverted operator $(M^j\circ)^{-1}$ in an analogous way to Sect. 4.4.1 the operator $M^j\circ$ is simplified by applying the assumptions made for the Fourier analysis in Chap. 4:

$$\begin{aligned} M^j\circ \;=\; & -\left(v\Delta x+\mu\frac{\Delta x}{\Delta y}\right)\mathcal{E}_j^{-1}\circ \\ & +\left(\frac{1}{\tau}+v\Delta x+2\mu\frac{\Delta x}{\Delta y}-\min\left(0,\mathcal{Z}_S\right)\right)\mathcal{I}\circ \\ & -\mu\frac{\Delta x}{\Delta y}\mathcal{E}_j\circ\;. \end{aligned} \tag{6.82}$$

Applying the j–line implicit operator to an arbitrary mode $\widehat{W}e^{\mathbf{I}i\Phi_x+\mathbf{I}j\Phi_y}$ reads

$$M^j\circ\,\mathcal{M}^j\cdot\widehat{W}e^{\mathbf{I}i\Phi_x+\mathbf{I}j\Phi_y} \;=\widehat{W}e^{\mathbf{I}i\Phi_x+\mathbf{I}j\Phi_y}\;. \tag{6.83}$$

Inserting the definition of $M^j\circ$, Eq. (6.82), to Eq. (6.83) and resolving for its Fourier symbol $\mathcal{M}^j$ yields

$$\begin{aligned} \mathcal{M}^j \;&=\; \frac{1}{\dfrac{1}{\tau}+v\Delta x\left(1-e^{-\mathbf{I}\Phi_y}\right)+\mu\dfrac{\Delta x}{\Delta y}\left(2-e^{-\mathbf{I}\Phi_y}-e^{\mathbf{I}\Phi_y}\right)-\min\left(0,\mathcal{Z}_S\right)} \\ &=\; \frac{1}{\dfrac{1}{\tau}-\mathcal{Z}_c|_j-\mathcal{Z}_v|_j-\min\left(0,\mathcal{Z}_S\right)}\,. \end{aligned} \tag{6.84}$$

$\mathcal{Z}_c|_j$ is part of Eq. (4.17) while $\mathcal{Z}_v|_j$ equals $\mathcal{Z}_v$ in Eq. (4.26). Instead of applying a Runge–Kutta method to

$$\frac{\partial\widehat{W}}{\partial t}e^{\mathbf{I}i\Phi_x+\mathbf{I}j\Phi_y} \;=\left(Z_c\circ\;+Z_v\circ\;+Z_S\circ\;\right)\widehat{W}e^{\mathbf{I}i\Phi_x+\mathbf{I}j\Phi_y} \tag{4.33}$$

as done in Sect. 4.3.1 here the j–line implicit scheme derived above is applied:

$$\widehat{W}^{n+1}e^{\mathbf{I}i\Phi_x+\mathbf{I}j\Phi_y} \;=\mathcal{G}_{j\text{–line}}\circ\,\widehat{W}^n e^{\mathbf{I}i\Phi_x+\mathbf{I}j\Phi_y}\;. \tag{6.85}$$

Elimination of $e^{\mathbf{I}i\Phi_x+\mathbf{I}j\Phi_y}$ yields

$$\widehat{W}^{n+1}=\left(1+\mathcal{M}^j\mathcal{Z}\right)\widehat{W}^n\,. \tag{6.86}$$

From Eq. (4.31) this results in

$$g_{j\text{–line}}=1+\mathcal{M}^j\mathcal{Z}\,, \tag{6.87}$$

which is Eq. (6.80) with operators replaced by their Fourier symbols, as the amplification factor of the j–line implicit scheme.

This amplification factor has been calculated for the model problem Eq. (4.1) with $S = 1000$, i. e. with a productive source term. The corresponding Fourier footprint is shown in Fig. 6.58. The footprint shows clearly that the amplification factor $g_{j\text{–line}}$ is independent of Φ_x. Furthermore the damping properties are very good for high– and even medium–frequency modes as $g_{j\text{–line}} < 0.3$ holds for any $|\Phi_y| > \pi/8$.
The CFL number taken for this calculation is $\texttt{CFL} = 150$ since varying this parameter yielded an optimum for the averaged damping properties at this value. This has been documented in Tab. 6.10. However, Tab. 6.10 shows amplification factor averages over

CFL number	$\overline{\lvert g_{j\text{–line}}\rvert}_{\text{all}}$	$\overline{\lvert g_{j\text{–line}}\rvert}_{\text{low}}$	$\lvert g_{j\text{–line}}\,(0,0)\rvert$
7.5	0.34375	0.55285	1.15
75.0	0.18543	0.34042	2.4997
150.0	0.17901	0.33442	3.9994
300.0	0.18594	0.35202	6.9988
1000.0	0.25271	0.4897	20.996

Table 6.10: Amplification factor averages and its absolute value for $(\Phi_x, \Phi_y) = (0, 0)$ related to the CFL number considering a productive source term coefficient ($S = 1000$)

all modes less than any other value of $\overline{|g|}_{\text{all}}$ obtained before. Even without a source term the Runge–Kutta method sped up with multigrid and implicit residual smoothing did not reach that small values as shown in Tab. 4.1. On the other hand $\overline{|g_{j\text{–line}}|}_{\text{low}}$ at $\texttt{CFL} = 150$ is about 40% smaller than $\overline{\left|g^{(q=5)}_{\text{pi}-\text{src},\,\text{IRS}}\right|}_{\text{low}}$ at any number of multigrid levels both applied to a productive source term with $S = 1000$. For this compare Tab. 5.7 on page 89.
At $\texttt{CFL} = 1000$ $|g_{j\text{–line}}\,(0,0)|$ has the same size as $\left|g^{(q=5)}_{\text{pi}-\text{src},\,\text{IRS}}\,(0,0)\right|$ at a four level V-cycle acting on the same productive source term. However, both $\overline{|g|}_{\text{all}}$ and $\overline{|g|}_{\text{low}}$ show significantly better damping properties of the j–line implicit method.

6.3.3.3 Computational Results with J–Line Implicit Scheme

The implementation of the j–line implicit method in the FLOWer code has been made straightforward: Along the lines of Mavriplis and Martinelli [80] the eddy viscosity provided by the turbulence model is calculated in front of the first Runge–Kutta stage on the finest grid only. Throughout a Runge–Kutta step as well as over a complete multigrid cycle the eddy viscosity is kept frozen. Thus the time integration scheme for the turbulence equations can be easily exchanged — considering the minimized coupling strategy, Sect. 6.2.1.2 — as long as it is done after the calculation and freezing of the eddy viscosity.
In order to take the same values of the mean quantities for both explicit and implicit time integration of the turbulence equations the implicit scheme is called between the eddy viscosity calculation and the call of the Runge–Kutta method for the RANS equations.

At first the j–line implicit scheme is applied to the RAE 2822 test case at Re $= 6.5 \cdot 10^6$ in combination with the improvements obtained in Sect. 6.1 and Sect. 6.2, i. e. singlegrid treatment and separate time step for the turbulence equations as well as minimized coupling. The variation of the CFL number done for the Fourier analysis in the section before has also been performed.
The convergence of the density residual is plotted in Fig. 6.59 followed by the convergence of the k residual in Fig. 6.60 and the convergence of the ω residual in Fig. 6.61. At CFLTU $= 7.5$ none of the residuals experience noteworthy speed–up compared to an explicit time integration as shown in Figs. 6.39 – 6.41. Increasing CFLTU up to 50 reduces the number of iterations necessary to drive the density residual to machine zero about 50%. The reduction of iterations for the k residual reaches $2/3$ while the ω residual only needs 40% of the iterations of the explicit Runge–Kutta method to reach machine accuracy.
The range of stable CFL numbers for the j–line approach for the turbulence equations has been tested by using CFLTU $= 1000$. The convergence of the density residual does not show considerable effects compared to the calculation using CFLTU $= 50$. Although the convergence rate of the k residual decreases at CFLTU $= 1000$ machine accuracy is reached in nearly the same number of iterations as at CFLTU $= 50$ since in that case the convergence rate changes after about seven orders of magnitude. Only for the ω residual a slow down in convergence has been observed at CFLTU $= 1000$. About one fifth of iterations more are needed compared to CFLTU $= 50$. Still the speed–up compared to the explicit treatment is about 50%. Thus there is no drawback in using CFLTU $= 1000$ as default for further investigations.

Applying the j–line implicit scheme with CFLTU $= 1000$ to a range of Reynolds numbers and correspondingly adapted grids for the RAE 2822 airfoil results in the convergence plots given in Figs. 6.62 – 6.64. Even at first glance it can be caught that the Re $= 100 \cdot 10^6$ calculation converges to machine accuracy for all residuals like the calculations for any other Reynolds number do. Additionally it needs to be pointed out that the speed–up observed for the calculations at Re $= 6.5 \cdot 10^6$ before is achieved or even beaten for all Reynolds numbers. This is most obvious for the convergence of the k residual in Fig. 6.63.

The line implicit treatment of the k–ω model raises the overall memory requirements of the FLOWer code by 2.5%. On the other hand the CPU time per iteration is reduced by 6% on a NEC SX–5 vector computer as the convective fluxes for the turbulence equations are calculated only once instead of five times (once per stage) and no implicit residual smoothing is performed for the turbulence equations. Thus only one tridiagonal system is solved instead of one per coordinate direction on every Runge–Kutta stage.

6.3.3.4 Conclusions for J–Line Implicit Scheme

As shown by Fourier analysis as well as by computational results the j–line implicit scheme is well suited for the time integration of turbulence equations. Although some optimum values of the corresponding CFL number have been detected by both Fourier analysis and computations the stability and efficiency of the method has been demonstrated for significantly larger values of CFLTU.

Applying the j–line implicit time integration to the turbulence equations speeds up the convergence of all residuals at minimum by 50% in terms of iterations necessary to reach machine accuracy. This speed–up is strengthened by a saving of 6% CPU time per iteration.

6.3.4 DDADI Scheme

A serious drawback of any line implicit approach is that only one coordinate direction can be treated implicitly. In case of three dimensional applications it may become difficult or even impossible to detect the coordinate direction best suited for an implicit treatment. Since the FLOWer code is used in industrial applications it is necessary to be able to treat 3D applications at a comparable level of robustness as 2D cases. Therefore the following sections deal with a three dimensional extension of the j–line implicit approach.

Further improvement of the approximation of the matrix M of the implicit scheme can be achieved by a more complex approximate factorization of M. The best known factorization approach is the ADI scheme originally introduced by Peaceman and Rachford [92].
For the implicit treatment of the turbulence equations a diagonal dominant variant of this scheme has been chosen. This DDADI scheme has been re–invented by MacCormack [76, 78, 77]. The scheme had been published by Bardina and Lombard [5], but its roots can be traced back to the early 60's [123]. It has also been found in the work of Venkatakrishnan [127], where the DDADI scheme has been employed as a preconditioner for an explicit Runge–Kutta scheme acting on the RANS equations.

The DDADI scheme for the turbulence equations is derived below. A Fourier analysis of this scheme has been performed and its results are presented. Conclusions are drawn after computational results using this implicit method have been discussed.

6.3.4.1 Derivation of DDADI Scheme

A line implicit scheme lacks generality for more complex test cases as the coordinate direction which should be treated implicitly might change throughout a grid: If there is a more or less perpendicular junction of two solid walls, only one of the two wall normal directions can be treated implicitly by means of a line implicit scheme. To overcome this shortcoming a fully implicit scheme seems favorable. Such a scheme yields in general a sparse matrix with a very large bandwidth. Instead of inverting this matrix directly, which is very costly, one tries to approximate it e. g. by a product of some easily to invert matrices. One possible choice for such a product is given by the diagonal dominant alternating direction implicit (DDADI) scheme [78, 77].

An ADI scheme is given if the matrix M is approximated by the product of matrices corresponding to line implicit treatment of each coordinate direction:

$$\mathsf{M} \approx \mathsf{M}^i \cdot \left(\frac{1}{\tau} \cdot \mathsf{I}\right)^{-1} \cdot \mathsf{M}^j \,. \tag{6.88}$$

I still denotes the identity matrix. The factor $([1/\tau] \cdot \mathsf{I})^{-1}$ is necessary to obtain an approximated but correctly scaled main diagonal for M. Otherwise

$$\mathsf{M}^i \cdot \mathsf{M}^j \cdot \overrightarrow{\Delta W} = \left(\frac{1}{\tau} \cdot \mathsf{I}\right)^2 \cdot \overrightarrow{\Delta W} + \ldots \tag{6.89}$$

would hold which uses a different, i. e. squared, time step compared to Eqs. (6.21) and (6.22), respectively.

The diagonal dominant ADI scheme is obtained if the main diagonals of M^i and M^j are taken the same, i. e.

$$\mathsf{M}^i = \mathsf{L}^i + \mathsf{D} + \mathsf{U}^i \tag{6.90}$$

for the i–line implicit contribution and

$$\mathsf{M}^j = \mathsf{L}^j + \mathsf{D} + \mathsf{U}^j \tag{6.91}$$

for the j–line part instead of Eq. (6.73). D is given as

$$\mathsf{D} = \frac{1}{\tau} \cdot \mathsf{I} - \mathsf{D}_c^i - \mathsf{D}_c^j - \mathsf{D}_v^i - \mathsf{D}_v^j - \mathsf{D}_S \,. \tag{6.92}$$

Therefore this main diagonal is independent of any coordinate direction. As the source term is taken into account for all directions the diagonal dominance of the line implicit matrices M^i and M^j is further increased. On the other hand it is no longer sufficient to divide the product of these matrices by the time step as in Eq. (6.88). As each coordinate direction contributes the same main diagonal, the product of M^i and M^j has to be multiplied by the inverse of D instead. Thus the matrix M of the full implicit scheme is approximated by the DDADI scheme as

$$\mathsf{M} \approx \mathsf{M}^i \cdot \mathsf{D}^{-1} \cdot \mathsf{M}^j \,, \tag{6.93}$$

a product of diagonal and tridiagonal matrices. As for the ADI scheme the correct scaling of the main diagonal of the approximated matrix M is obtained by multiplying by D^{-1}. The resulting scheme

$$\mathsf{M}^i \cdot \mathsf{D}^{-1} \cdot \mathsf{M}^j \cdot \overrightarrow{\Delta W} \approx Z \circ \overrightarrow{W^n} \tag{6.94}$$

approximates Eq. (6.22). Each product is solved by means of the Thomas' algorithm for tridiagonal systems independent of its bandwidth or directly if possible. The system is solved in three steps:

1. Solve the tridiagonal system
$$\mathsf{M}^i \cdot \overrightarrow{\Delta W^*} = Z \circ \overrightarrow{W^n} \tag{6.95}$$
by means of the Thomas' algorithm.

2. Calculate directly
$$\mathsf{D}^{-1} \overrightarrow{\Delta W^{**}} = \overrightarrow{\Delta W^*} \tag{6.96}$$
as D is a diagonal matrix.

3. Solve the tridiagonal system
$$\mathsf{M}^j \cdot \overrightarrow{\Delta W} = \overrightarrow{\Delta W^{**}} \tag{6.97}$$
by means of the Thomas' algorithm.

The extension to three dimensions is straightforward.

6.3.4.2 Fourier Analysis of DDADI Scheme

The operator notation necessary for the Fourier analysis in this section reads for a single point

$$M^i\circ\, D^{-1}\circ\, M^j\circ\, W_{i,j}^{n+1} \approx \left(M^i\circ\, D^{-1}\circ\, M^j\circ\, + Z\circ\right) W_{i,j}^{n} \tag{6.98}$$

i. e.

$$W_{i,j}^{n+1} \approx \mathcal{G}_{\mathrm{DDADI}}\circ\, W_{i,j}^{n} \tag{6.99}$$

with the operator for the DDADI time stepping scheme

$$\mathcal{G}_{\mathrm{DDADI}}\circ\, = \mathcal{I}\circ\, + \left(M^j\circ\right)^{-1} D\circ\, \left(M^i\circ\right)^{-1} Z\circ\, . \tag{6.100}$$

The definitions of $M^i\circ$, $M^j\circ$ and $D\circ$ are summarized from previous sections as

$$\begin{aligned} M^i\circ\, \;=\; & \left(-\, Z_c|^i\Big[_{i,j}^{\mathcal{L}} - Z_v|^i\Big[_{i,j}^{\mathcal{L}}\right)\mathcal{E}_i^{-1}\circ \\ & + \left(\frac{1}{\tau} - Z_c|^i\Big[_{i,j}^{\mathcal{D}} - Z_c|^j\Big[_{i,j}^{\mathcal{D}} - Z_v|^i\Big[_{i,j}^{\mathcal{D}} - Z_v|^j\Big[_{i,j}^{\mathcal{D}} - Z_S|^j\Big[_{i,j}^{\mathcal{D}}\right)\mathcal{I}\circ \\ & + \left(-\, Z_c|^i\Big[_{i,j}^{\mathcal{U}} - Z_v|^i\Big[_{i,j}^{\mathcal{U}}\right)\mathcal{E}_i\circ\, , \end{aligned} \tag{6.101}$$

$$\begin{aligned} M^j\circ\, \;=\; & \left(-\, Z_c|^j\Big[_{i,j}^{\mathcal{L}} - Z_v|^j\Big[_{i,j}^{\mathcal{L}}\right)\mathcal{E}_j^{-1}\circ \\ & + \left(\frac{1}{\tau} - Z_c|^i\Big[_{i,j}^{\mathcal{D}} - Z_c|^j\Big[_{i,j}^{\mathcal{D}} - Z_v|^i\Big[_{i,j}^{\mathcal{D}} - Z_v|^j\Big[_{i,j}^{\mathcal{D}} - Z_S|^j\Big[_{i,j}^{\mathcal{D}}\right)\mathcal{I}\circ \\ & + \left(-\, Z_c|^j\Big[_{i,j}^{\mathcal{U}} - Z_v|^j\Big[_{i,j}^{\mathcal{U}}\right)\mathcal{E}_j\circ \end{aligned} \tag{6.102}$$

and

$$D\circ\, = \left(\frac{1}{\tau} - Z_c|^i\Big[_{i,j}^{\mathcal{D}} - Z_c|^j\Big[_{i,j}^{\mathcal{D}} - Z_v|^i\Big[_{i,j}^{\mathcal{D}} - Z_v|^j\Big[_{i,j}^{\mathcal{D}}\right)\mathcal{I}\circ\, . \tag{6.103}$$

The contributions of viscous terms in i–direction, $Z_v|^i\Big[_{i,j}^{*}$, have been taken into account only for the notation of these matrices. In the framework of thin layer approximation as used here they are neglected. Together with this approximation the simplification by the assumptions made for the Fourier analysis yields

$$\begin{aligned} M^i\circ\, \;=\; & -u\Delta y\,\mathcal{E}_i^{-1}\circ \\ & + \left(\frac{1}{\tau} + u\Delta y + v\Delta x + 2\mu\frac{\Delta x}{\Delta y} - \min\left(0, \mathcal{Z}_S\right)\right)\mathcal{I}\circ \\ & - 0\,\mathcal{E}_i\circ\, . \end{aligned} \tag{6.104}$$

The operator $M^j\circ$ is no longer defined by Eq. (6.82) but reads

$$\begin{aligned} M^j\circ\, \;=\; & -\left(v\Delta x + \mu\frac{\Delta x}{\Delta y}\right)\mathcal{E}_j^{-1}\circ \\ & + \left(\frac{1}{\tau} + u\Delta y + v\Delta x + 2\mu\frac{\Delta x}{\Delta y} - \min\left(0, \mathcal{Z}_S\right)\right)\mathcal{I}\circ \\ & - \mu\frac{\Delta x}{\Delta y}\,\mathcal{E}_j\circ\, . \end{aligned} \tag{6.105}$$

The only difference lies in the contribution of $u\Delta y$ to the main diagonal. The operator of the main diagonal reads

$$D\circ = \left(\frac{1}{\tau} + u\Delta y + v\Delta x + 2\mu\frac{\Delta x}{\Delta y} - \min\left(0, \mathcal{Z}_S\right)\right)\mathcal{I}\circ \ . \tag{6.106}$$

As the complete system is solved in three steps, for each operator its Fourier symbol is obtained independent of the others. In analogy to the derivation of the Fourier symbol of the j–line implicit operator the Fourier symbol $\mathcal{M}^i$ is derived as

$$\begin{aligned}\mathcal{M}^i &= \frac{1}{\frac{1}{\tau} + u\Delta y\left(1 - e^{-\mathbf{I}\Phi_x}\right) + v\Delta x + 2\mu\frac{\Delta x}{\Delta y} - \min\left(0, \mathcal{Z}_S\right)} \\ &= \frac{1}{\frac{1}{\tau} - \mathcal{Z}_c|_i + v\Delta y + 2\mu\frac{\Delta x}{\Delta y} - \min\left(0, \mathcal{Z}_S\right)} \ .\end{aligned} \tag{6.107}$$

The Fourier symbol $\mathcal{M}^j$ is nearly identical to the Fourier symbol of the j–line implicit in Eq. (6.84) except for the contribution of the convection in i–direction, $u\Delta y$:

$$\mathcal{M}^j = \frac{1}{\frac{1}{\tau} + u\Delta y - \mathcal{Z}_c|_j - \mathcal{Z}_v|_j - \min\left(0, \mathcal{Z}_S\right)} \ . \tag{6.108}$$

The Fourier symbol $\mathcal{D}$ of the diagonal operator $D\circ$ is derived by inserting its definition, Eq. (6.106), to

$$\mathcal{D}\cdot\widehat{W}e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y} = D\circ\widehat{W}e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y} \ . \tag{6.109}$$

Resolving for $\mathcal{D}$ yields

$$\mathcal{D} = \frac{1}{\tau} + u\Delta y + v\Delta x + 2\mu\frac{\Delta x}{\Delta y} - \min\left(0, \mathcal{Z}_S\right) \ . \tag{6.110}$$

The amplification factor g_{DDADI} for the DDADI scheme is derived from the application of the corresponding operator $\mathcal{G}_{\mathrm{DDADI}}\circ$ as a time stepping scheme

$$\widehat{W}^{n+1}e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y} = \mathcal{G}_{\mathrm{DDADI}}\circ\widehat{W}^{n}e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y} \ . \tag{6.111}$$

Again $e^{\mathbf{I}i\Phi_x + \mathbf{I}j\Phi_y}$ is eliminated and Eq. (4.31) is applied yielding

$$g_{\mathrm{DDADI}} = 1 + \mathcal{M}^j\mathcal{D}\mathcal{M}^i\mathcal{Z} \ . \tag{6.112}$$

This equation corresponds to Eq. (6.100) but with Fourier symbols instead of operators.

In Fig. 6.65 the Fourier footprint of g_{DDADI} is plotted at $\mathtt{CFL} = 150$ with $S = 1000$. No difference can be detected compared to the Fourier footprint of $g_{j\text{–line}}$ in Fig. 6.58. This is due to the fact that the additional influence of the i–direction is nearly negligible as it scales with $u\Delta y = 10^{-7}$. On the other hand the contributions of the j–line treatment scale with $v\Delta x + 2\mu\Delta x/\Delta y \approx 10^{-4}$.
In Tab. 6.11 the averages of the amplification factor are listed. In comparison to Tab. 6.10 the averages are slightly increased at the lower CFL numbers while they are improved at higher CFL numbers. Overall no significant difference in the averaged amplification

CFL number	$\overline{\lvert g_{\mathrm{DDADI}}\rvert}_{\mathrm{all}}$	$\overline{\lvert g_{\mathrm{DDADI}}\rvert}_{\mathrm{low}}$	$\lvert g_{\mathrm{DDADI}}(0,0)\rvert$
7.5	0.34394	0.55301	1.1498
75.0	0.18572	0.34071	2.478
150.0	0.17902	0.33413	3.913
300.0	0.18476	0.34934	6.6612
1000.0	0.23548	0.45456	17.6694

Table 6.11: Amplification factor averages and its absolute value for $(\Phi_x, \Phi_y) = (0, 0)$ related to the CFL number considering a productive source term coefficient ($S = 1000$) — singlegrid analysis

factors can be seen from these tables.
The absolute value of $g_{\mathrm{DDADI}}(0,0)$ on the other hand differs more and more from its counterpart for the j–line implicit scheme with increasing CFL number. At $\mathtt{CFL} = 1000$ $|g_{\mathrm{DDADI}}(0,0)|$ is about 15% less than $|g_{j\text{–line}}(0,0)|$. This difference is due to the coordinate direction independent diagonal matrix $\mathtt{D}$ used within the DDADI scheme. As its entries are independent of any Fourier mode its effect directly scales with the CFL number. At least this is the same effect as with the point implicit treatment of the whole model equation.

6.3.4.3 Computational Results with DDADI Scheme

The same computations as for the j–line scheme have been performed for the DDADI scheme.
Figs. 6.66 – 6.68 show that the variation of CFLTU results in an improved convergence for all residuals especially in the first 500 iterations. While the k residual experiences improvements at higher CFL numbers for the implicit scheme — at $\mathtt{CFLTU} = 1000$ machine accuracy is reached in about 200 cycles less — the ω residual is sped up over the whole range of CFL numbers. However, the convergence of the ω residual is also improved the most at $\mathtt{CFLTU} = 1000$.

Over the range of Reynolds numbers the convergence of all residuals is improved compared to the j–line implicit treatment of the turbulence equations as shown in Figs. 6.69 – 6.71. These improvements are not as significant as those obtained by the j–line implicit scheme compared to the explicit Runge–Kutta time stepping scheme as discussed in Sect. 6.3.3.3. The improvements of the DDADI scheme over the j–line implicit scheme require 2% additional CPU time when using the DDADI scheme instead of its line implicit counterpart on a NEC SX–5 vector computer. Additionally DDADI requires about 3.5% more memory. Compared to the Runge–Kutta scheme 4% in CPU time are still saved by the application of the DDADI scheme. These savings in CPU time are founded on the same reasons as the savings obtained by the line implicit scheme.

6.3.4.4 Conclusions for DDADI Scheme

Both Fourier analysis and computational results showed the satisfying performance of the DDADI scheme applied to the turbulence equations. As for the j–line implicit scheme some optimum values of the corresponding CFL number have been detected by Fourier analysis as well as by computations. However, the stability and efficiency of the DDADI scheme has been demonstrated for significantly larger values of `CFLTU`. Furthermore, the CPU time needed for the DDADI scheme does not counteract the improvements obtained in terms of iterations compared to the j–line scheme even for two dimensional applications. Since the DDADI scheme takes into account all coordinate directions it is more suited for 3D applications than the j–line implicit scheme.

6.3.5 Implicit Treatment of Boundary Conditions

In case of implicit schemes used for the integration of differential equations the treatment of the corresponding boundary conditions has to be considered.
At first it must be pointed out that the implicit treatment of boundary conditions for the turbulence variables which are independent of the turbulence quantities themselves (e. g. at wall and farfield, respectively) is not possible. Thus the only boundaries which might be treated implicitly are cuts (interface boundaries between blocks). Their implicit treatment is complicated and is a research topic in its own right (cmp. e. g. LeChuiton [23]).

Two–dimensional test cases have been performed to clarify the influence of cut boundaries on the convergence behaviour of the implicitly treated quantities. For these computations the Reynolds number adapted grid at $\mathrm{Re} = 6.5 \cdot 10^6$ has been split into 2, 4 and 16 blocks. The block structures are shown in Figs. 6.72 – 6.74. The splitting of the grid has been done in such a way that the same multigrid capability — 4 level multigrid — as for the single block grid is assured: At first 40 grid cells adjacent to the airfoil resp. to the wake cut are taken as a "boundary layer block" while the outer region builds a "farfield block" as shown in Fig. 6.72. This outer region has been cut into one block in front of the nose and two others on the upper and lower side of the airfoil, Fig. 6.73. Each of the resulting 4 blocks has been cut into four parts resulting in the 16 block structure shown in Fig. 6.74. Care has been taken that no further cuts end at the trailing edge as this may give raise to severe problems for the cell vertex metric in use.

On these grids calculations have been performed using the DDADI scheme for the turbulence equations with the same settings as for the Reynolds number variations in Sect. 6.3.4.3. The convergence behaviour of the density residual is plotted in Fig. 6.75. Except for the 16 block case no difference in convergence can be seen. The identical observation is made for the convergence of the k residual — Fig. 6.76 — and the ω residual in Fig. 6.77.
For the 16 block case the convergence behaviour of all residuals is slightly influenced but only minimized coupling between the RANS and the turbulence equations has been applied. Thus the effect on the density residual is not introduced by the turbulence quantities. As both mean and turbulence equations are treated implicitly — the latter are integrated by means of the DDADI scheme while the first are subject to

implicit residual smoothing — the observed change in convergence is based on the explicit treatment of cuts for both implicit approaches. However, as the implicit residual smoothing for the mean equations yields the same effect on convergence in presence of cut boundaries as the implicit treatment of the turbulence equations no necessity arises to introduce an implicit treatment of the cut boundaries specifically for the turbulence equations.

6.3.6 Conclusions from Implicit Treatment of Turbulence Equations

Three implicit schemes have been introduced and implemented for the turbulence equations: A point implicit treatment of the complete turbulence equation, a line implicit approach and a diagonal dominant alternating direction implicit (DDADI) scheme. Both Fourier analysis and computations have been used to investigate the performance of the different approaches. The most efficient damping properties have been predicted by Fourier analysis for the line implicit and the DDADI scheme at an optimum CFL number. Best computational results in terms of iterations have been obtained by performing a standard multigrid cycle, i. e. four level W–cycle, for the mean flow equations but a DDADI singlegrid calculation for the turbulence equations. The overall CPU time needed to reach machine zero has also been lowest for this combination. Besides singlegrid treatment other improvements obtained in previous sections — a separate time step as well as a minimized coupling — have been applied to the turbulence equations.

For engineering applications the increase in robustness by applying the line implicit or the DDADI approach to the turbulence equations is more relevant:
Convergence to an averaged density residual of about 10^{-5} is expected to result in a steady state solution in general. If the numerical simulation does not show robustness problems like for the RAE 2822 airfoil at Re $= 6.5 \cdot 10^6$ the efficiency of the computation is reduced due to the implicit singlegrid treatment of the turbulence equations. For convenience the corresponding data have been combined in Figs. 6.78 – 6.81. The convergence of the density residual for different time integration approaches of the turbulence equations is shown in Fig. 6.78. The highest efficiency is observed for the explicit multigrid treatment of the turbulence equations. About 200 multigrid cycles are needed to reach a convergence of 5 orders of magnitude while with the second fastest approach, the DDADI scheme, about 300 iterations are necessary. For the turbulence equations — Figs. 6.79 and 6.80 — the situation is not as clear but they are usually not taken as crucial convergence criteria in engineering applications. Furthermore, the convergence behaviour of the lift coefficient, Fig. 6.81, is the same as for the density residual: After about 200 multigrid cycles a steady value is reached on the basis of multigrid treated turbulence equations. 50% more iterations are needed when applying the DDADI scheme to the turbulence equations.
However, at higher Reynolds numbers with application of the DDADI scheme to the turbulence equations a convergence level of about 10^{-5} for the density residual is reached within the same number of iterations (cmp. Fig. 6.69). On the other hand the density residual fails to reach the same level of convergence in case of an explicit multigrid treatment of the turbulence equations at higher Reynolds numbers as shown in Fig. 5.17.

A remark on the aerodynamic coefficients for lift and drag is necessary: As already pointed out in Sect. 5.1.3.1 the multigrid treatment of the turbulence equations influences the lift and the drag coefficient. In Tab. 6.12 these aerodynamic coefficients are given for different time integration approaches applied to the turbulence equations. It is obvious from this table that the time integration scheme — explicit, line implicit

time integration	lift	drag
RK + SG	0.798361	0.018146
RK + W4	0.798727	0.018162
j–line + SG	0.798355	0.0181463
DDADI + SG	0.798352	0.0181462

Table 6.12: Aerodynamic coefficients for lift and drag related to the time integration approach applied to the k–ω turbulence model (RANS : RK + W4); RAE 2822 airfoil at Re $= 6.5 \cdot 10^6$

or full implicit — does considerably change neither lift nor drag. The differences in the coefficients are two orders of magnitude smaller than those introduced by multigrid treatment of turbulence equations. Fig. 6.82 shows the distribution of the pressure coefficient for the same computation which are listed in Tab. 6.12 according to their aerodynamic coefficients. Any differences are below graphical accuracy. Therefore a close–up of the C_p distribution is given in Fig. 6.83. It is a strong zoom on the upper part of the shock region. While nearly no differences are visible for any time integration approach using singlegrid for the turbulence equations, the solution of the computation with multigrid treated turbulence equations shows a small but distinctive offset. This offset has been found in large regions of the C_p distribution. A comparable offset has been found in the distribution of the skin friction coefficient C_f.
From these results it has to be concluded that the computed boundary layer is slightly changed if the turbulence equations are treated with multigrid.

Due to its restriction to one coordinate direction the j–line implicit approach is applicable to simple geometries and block structures only. For more complex geometries and block structures the DDADI scheme with its capability to treat all coordinate directions implicitly has a clear advantage. This higher flexibility and the gain in6 robustness will be demonstrated in Chap. 7.

6.4 Conclusions

In Sect. 6.1 it has been shown that multigrid treatment of a productive source term counteracts the improvements in damping of low–frequency error modes usually gained by multigrid. This is due to the problem that both source term and multigrid affect low–frequency modes. In the same way as damping of low–frequency modes e. g. by a Runge–Kutta scheme is intensified by multigrid the amplification of low–frequency modes due to a productive source is intensified by multigrid as well. Thus a singlegrid treatment for the turbulence equations has been deduced to be most robust.

The coupling between the RANS and the turbulence equations has been investigated in Sect. 6.2.1. A minimized coupling via the eddy viscosity has been found by computational experiments to be most robust.
A higher efficiency has been obtained by a separate time step for the turbulence equations in Sect. 6.2.2. This time step is enlarged compared to the time step used for time integration of the mean equations. It is based on the fact that turbulence quantities do not experience information transport by acoustic waves as RANS quantities do.

A more implicit time integration of the turbulence equations has been investigated in Sect. 6.3. While an extended point implicit treatment raised the CPU time per iteration neither efficiency nor robustness were found to be improved by this approach. By Fourier analysis and computations an extensive improvement in robustness and efficiency has been demonstrated for a line implicit treatment of the turbulence equations in the wall normal direction. The efficiency has been found to be raised in terms of iterations as well as in CPU time per iteration. Further reduction of the number of iterations needed to reach machine accuracy has been obtained by the application of a diagonal dominant alternating direction implicit (DDADI) scheme. The memory requirements for these implicit schemes raised the overall memory used by the FLOWer code by less than 7%.

The combination of all these improvements — time integration for the turbulence equations using the DDADI scheme with a separate time step on a singlegrid basis without modifying the time integration of the RANS equations and coupling only via the eddy viscosity to the RANS equations — has been found to yield sufficiently high robustness for simulations of turbulent flow around two dimensional airfoils at flight Reynolds numbers using transport equation turbulence models. Besides increased robustness this combination of improvements also sped up the computations performed by about 50% in terms of iterations and additional 4% in CPU time per iteration.

7 Applications

In this chapter the main application for which also Reynolds number variations have been performed is the ALVAST wing/body configuration of DLR [58]. The corresponding calculations are discussed in Sect. 7.1. The robustness gained by application of the DDADI scheme is demonstrated in subsequent sections for two more test cases. Results for the ALVAST wing/body configuration with deployed high lift devices are presented in Sect. 7.2. The robustness enhancement due to the implicit treatment of the turbulence equations for 2D cases is demonstrated by some results for a helicopter rotor blade profile in Sect. 7.3.

7.1 ALVAST Wing/Body Configuration

In this section results for the ALVAST wing/body configuration [58] over a range of Reynolds numbers are presented. The Reynolds number is varied from wind tunnel to flight Reynolds number and beyond. Besides the wind tunnel Reynolds number $\mathrm{Re} = 4.3 \cdot 10^6$ three larger Reynolds numbers have been chosen. As the ALVAST wind tunnel model is scaled by a factor of about 10 compared to its realistic counterpart [58], $\mathrm{Re} = 43 \cdot 10^6$ has been chosen as high resp. flight Reynolds number for this model. Additionally two more Reynolds numbers — $\mathrm{Re} = 86 \cdot 10^6$ and $\mathrm{Re} = 129 \cdot 10^6$ — are considered in order to check the improvements for even higher Reynolds numbers.

For all these Reynolds numbers 3D multiblock grids have been generated by means of the grid generation tool MegaCads [17]. As discussed for the RAE 2822 airfoil, Sect. 5.1.1, an adaptation of the boundary layer resolution related to the Reynolds number is necessary. Thus the grids have been adapted to the Reynolds number as described in Sect. 7.1.1.

For the computational results on these Reynolds number adapted grids flow conditions have been taken as $M_\infty = 0.75$ and $\alpha = 1.0^\circ$ as given in [20]. The results obtained by the baseline FLOWer code as well as by its improved version — with emphasis on the convergence behaviour — are discussed in Sect. 7.1.2.

Sect. 7.1.3 closes this section by some conclusions.

7.1.1 3D Grid Generation for High Reynolds Number Flows

In this section the grids generated for the 3D wing/body configuration ALVAST are described.

For each Reynolds number specified above the boundary layer resolution of the corresponding grid has been adapted as for the RAE 2822 airfoil (Sect. 5.1.1). For this purpose at three spanwise positions on the wing — wing/fuselage intersection, wing kink ($\eta \approx 0.36$) and wing tip ($\eta \approx 0.85$) — two dimensional grids normal to the wing surface have been generated in analogy to the basic grid construction for the RAE 2822 airfoil. These grids contain several supports to approximate the boundary layer thickness. Additionally at the same supports the first spacing normal to the wall can be adjusted. For the resolution of the boundary layer 24 points are available. The wing profile is resolved by 272 grid points while the spanwise direction is discretized by means of 96 grid points.
Based on these boundary layer adaptable grids a 3D CO–grid around the wing of the ALVAST configuration has been generated. The O–type grid is in spanwise direction while the C–type grid is perpendicular to it. Furthermore, the fuselage has been covered by a grid which has been boundary layer adapted according to flat plate boundary layer approximations. The basic MegaCads control file has been generated by M. Rakowitz for the DLR F4 wing/body configuration [99]. As DLR F4 and ALVAST are quite similar [58] only some adaptations have been necessary. Additionally extensions to the objective to generate boundary layer adapted grids for high Reynolds number flows have been introduced to the MegaCads control file. Especially the generation of grid cells with non–negative volumes on the upper side of the wing in the vicinity of the wing/fuselage junction has been a non trivial task for the larger Reynolds numbers.
The grid generation resulted in Re–adapted grids each consisting of seven blocks with about 4.3 million grid points in total.

Adaptation of the boundary layer resolution has been made more or less by hand for two reasons: On the one hand the Baldwin/Lomax turbulence model [4] is not well suited for multiblock computations. On the other hand running two to three sufficiently converged computations for each Reynolds number just to adapt the boundary layer thickness and the first spacing on grids containing about 4.3 million points has been regarded as not possible (in terms of CPU time) for an engineering application.
Thus the experiences gained from the boundary layer adaptation of the grids for the RAE 2822 airfoil (Sect. 5.1.1) have been used to adapt the boundary layer of the ALVAST wing. As noted by Tassa et al. [116] the boundary layer thickness in 3D tends to be thicker close to the trailing edge than in 2D. Thus a boundary layer thickness as for the RAE 2822 airfoil at $\mathrm{Re} = 6.5 \cdot 10^6$ has been supposed to be a suitable, probably underestimated approximation at $\mathrm{Re} = 4.3 \cdot 10^6$. The first spacing has been scaled by the ratio of these Reynolds numbers as y^+ can be interpreted as a local Reynolds number [119]. The grid at $\mathrm{Re} = 4.3 \cdot 10^6$ has been checked for its boundary resolution based on these approximations. In a second step these approximated parameters have been scaled based on the ratios of the corresponding values for the RAE 2822 airfoil at different Reynolds numbers in order to get settings for the boundary layer adaptation at the larger Reynolds numbers.

Figs. 7.1 – 7.4 show y_w^+ distributions for all Reynolds numbers investigated. As it can be seen y_w^+ is less than 2 nearly everywhere on the surfaces. In general it is not larger than 5 for all Reynolds numbers.
This also holds for the inner wing area at $\mathrm{Re} = 129 \cdot 10^6$, Fig. 7.4. Here grid generation became more difficult due to very small spacings in combination with the concave ge-

ometry shape of the fuselage wing intersection. Since two wall normal directions have to be considered in this region wall normal spacings necessary to approach $y_w^+ \approx 1$ yielded cells with "negative" volume. If the computation of the volume of a grid cell gives a negative value it is a typical sign for highly distorted, skewed grid cells. Increasing the corresponding spacings often allows grid generation without "negative" volumes. Here the spacings in wall normal direction for the fuselage have been increased around the fuselage wing intersection. Due to the block structure applied y_w^+ on the fuselage behind the wing is also enlarged by this increment. The same effect leads to the enlarged y_w^+ values on the upper wing surface as the wall normal spacing of the wing has been increased at the fuselage wing intersection. The improved y_w^+ distribution on the rest of the wing and the fuselage compared to its distribution at other Reynolds numbers results from the grid generation problems, too. Because of the necessity to adjust parameters much more carefully to avoid distorted cells in some regions the parameter settings for the other areas got improved additionally.
One exception from the upper bound of the y_w^+ values has to be made for the outermost tip area at all Reynolds numbers. However, this region is negligibly small.

Fig. 7.5 demonstrates the quality of boundary layer resolution over all Reynolds numbers. These dimensionless velocity profiles are based on the same formulae as in Sect. 5.1.1.3. While the laminar sublayer is not as well resolved as for the RAE 2822 airfoil (Fig. 5.5) the slope of the velocity profile fits quite well to the logarithmic law of wall for all Reynolds numbers except the wind tunnel Reynolds number $\mathrm{Re} = 4.3 \cdot 10^6$. This is the same as observed for the boundary layer resolution of the RAE 2822 airfoil. Although Fig. 7.5 shows data of only one surface point the boundary layer resolution has been checked at other points as well yielding comparable results.

Considering the reduced number of grid points inside the approximated boundary layer — about 20 instead of 30 — and the higher complexity of 3D grid generation the quality of these Re–adapted grids related to the resolution of the boundary layer is satisfying.

7.1.2 Computational Results for High Reynolds Number Flows

In this section results obtained on the Re–adapted grids described in the previous section are presented. Emphasis is laid on the discussion of the convergence behaviour. Since the investigation of the variation of FLOWer input parameters for a 3D test case like this is a quite costly approach one set of parameters has been chosen for all computations except for those parameters introduced in the framework of this work. Thus the potential benefit due to the improved treatment of the turbulence equations in terms of increased stability for the RANS equations (e. g. using a larger CFL number) has not been investigated. Hence, all computations have been performed with the same set of input parameters except for the Reynolds number and the switch for the time stepping scheme for the turbulence equations. An automatic default setting for the CFL number for the time integration of the turbulence equations, `CFLTU`, as introduced below reduces the required user input further.

The RANS equations are treated by a three level W–cycle embedded in a full multigrid (FMG) approach starting on the third level. Thus only singlegrid is used on the coarsest level while on the medium level a two level cycle is employed. The whole three

level W–cycle is only used on the finest grid. 1000 iterations on each coarse level are performed followed by 2000 iterations on the finest grid. $\texttt{CFL} = 5.5$ has been used. For the CFL number of the time integration of the turbulence equations, CFLTU, an automatic default setting has been implemented in the FLOWer code: In case of baseline treatment of turbulence equations — identical time integration as for RANS equations — CFLTU is chosen to equal CFL. When applying the implicit time integration to the k–ω equations in combination with singlegrid the setting of CFLTU is made by the following rule of thumb:
In order to balance the speed–up of the time integration of the RANS equations by means of multigrid CFLTU is chosen $N \cdot \texttt{CFL}$ with N denoting the number of singlegrid time steps necessary to reach the same time evolution as with the multigrid cycle in use. Based on the assumption of doubling the time step size when switching to the next coarser grid $N = 1 + 2 = 3$ for a two level cycle, i. e. one singlegrid time step and one coarse grid time step which equals two singlegrid time steps. For a sawtooth V–cycle N is calculated as

$$N = 2^{\texttt{levels}} - 1 \tag{7.1}$$

for any number of multigrid levels in use (levels). For a sawtooth W–cycle the formula is more complex and reads

$$N = \max\left(\left[\sum_{i=0}^{\texttt{levels}} 4^i - 2 \cdot 4^{\texttt{levels}-1}\right], 1\right) . \tag{7.2}$$

Related to a three level W–cycle $N = 13$. Thus $\texttt{CFLTU} = 71.5$ for the implicit time integration of the turbulence equations on the finest grid level for the computations presented here.

The convergence results obtained by the baseline FLOWer code are shown in Figs. 7.6 – 7.8. At first the absence of a convergence curve at $\text{Re} = 129 \cdot 10^6$ needs to explained. It started to converge on the two coarser grid levels of the full multigrid computation. After transferring data to the finest grid level the computation diverged within 10 iterations. The reason for this divergence will be discussed in more detail when the results for the improved version of FLOWer are presented.
The density residual converges over the complete full multigrid computation as shown in Fig. 7.6. However, on the finest grid the convergence behaviour becomes more perturbed. Convergence to 10^{-4} resp. 10^{-5} (at $\text{Re} = 4.3 \cdot 10^6$) is obtained for all Reynolds numbers except for $\text{Re} = 129 \cdot 10^6$. The convergence of the k residual is presented in Fig. 7.7. While a transient phase is passed on the coarsest level the convergence breaks down on both finer grids. Remember that on these two grid levels multigrid cycles are performed. At $\text{Re} = 86 \cdot 10^6$ the k residual diverges slightly on the finest grid level. This slope may change when continuing the computation. For the ω residual, Fig. 7.8, the convergence is smooth on coarser grids while it shows significant perturbations on the finest grid level.

The application of the implicit time integration, i. e. DDADI scheme, to the turbulence equations together with a singlegrid approach and minimized coupling results in convergence as shown in Figs. 7.9 – 7.12. Comparing the convergence of the density residual in Fig. 7.9 to Fig. 7.6 three differences are remarkable. In the first place convergence of the density residual is obtained at $\text{Re} = 129 \cdot 10^6$. Secondly the convergence on the finest grid level is smooth for all Reynolds numbers. And — like for the RAE

2822 airfoil at Re $= 6.5 \cdot 10^6$ — the convergence of the density residual at Re $= 4.3 \cdot 10^6$ is less efficient compared to the baseline method.
The first two remarks hold also for the k residual as shown in Fig. 7.10 compared to Fig. 7.7. Although the same value for CFLTU is used on the first starting level of the full multigrid computation the transient phase is passed much faster — less than a quarter of iterations is needed — in case of applying the DDADI scheme to the turbulence equations. The convergence behaviour of the k residual on the finer grid levels can not be interpreted uniquely. However, on both grids smoothness of convergence is still given. But convergence behaves differently for each Reynolds number even compared to Fig. 7.7.
The convergence of the ω residual is plotted in Figs. 7.11 and 7.12. In Fig. 7.11 it can be seen that the convergence behaviour is significantly improved by application of the DDADI scheme to the turbulence equations except for some oscillations on the coarsest level. It has been found that these oscillations are due to the limitation of the turbulence quantities. The reason for the second convergence plot lies in the absence of the convergence curve at Re $= 129 \cdot 10^6$ on the finest grid level. Looking at Fig. 7.12 it is seen that the ω residual experiences a jump over 25 orders of magnitude within the first iteration on the finest grid level. However, the DDADI scheme is stable enough to overcome this jump and converges rapidly — within 50 iterations — to more reasonable residual values. There are two reasons why no visible effects on the other residuals can be detected. The minimized coupling prevents the density residual to be influenced. (An enlarged ω leads to a reduced eddy viscosity at maximum.) The k residual does not show any effect as the production of k is limited according to Menter [85] by 5 times its destruction.
An analogous behaviour destabilizes the baseline treatment of the turbulence equations so much that it diverges as noted above. Here the boosting of the ω residual experiences a further amplification due to multigrid. The destabilization of k is due to multigrid, too. Since the standard coupling between the RANS and the turbulence equations is used the density residual experiences the same destabilization as the k residual. The corresponding residual values are given in Tab. 7.1. As noted in Sect. 5.1.3.3 the very first residual is always calculated based on a singlegrid iteration. Multigrid effects are enclosed from the second iteration on.

level no.	iteration no.	ρ residual	k residual	ω residual
2	1000	$0.163 \cdot 10^1$	$0.175 \cdot 10^1$	$0.155 \cdot 10^8$
1	1	$0.181 \cdot 10^4$	$0.193 \cdot 10^4$	$0.210 \cdot 10^{33}$
1	2	$0.110 \cdot 10^{27}$	$0.827 \cdot 10^{24}$	$0.119 \cdot 10^{56}$
1	3	$0.160 \cdot 10^{27}$	$0.374 \cdot 10^{23}$	$0.973 \cdot 10^{55}$
1	4	$0.832 \cdot 10^{26}$	$0.233 \cdot 10^{22}$	$0.952 \cdot 10^{55}$

Table 7.1: Averaged, but non normalized residual values of the first fine grid iterations with original FLOWer treatment of turbulence equations for the ALVAST wing/body configuration at Re $= 129 \cdot 10^6$

Overall the convergence for all residuals at all Reynolds numbers has become smoother and more robust when applying the improved numerical treatment to the turbulence equations.

In Fig. 7.13 the convergence of the aerodynamic coefficients for lift, (total) drag and the friction related part of the drag is shown resulting from the calculations performed at $\mathrm{Re} = 4.3 \cdot 10^6$. The friction drag coefficient is taken because it is closely related to the convergence of the flow quantities within the boundary layer. On the coarser levels the speed–up of convergence for all aerodynamic coefficients due to the improved numerical treatment of the turbulence equations can be seen clearly. On the finest grid a visible speed–up is given only for the friction drag coefficient.
The distribution of the pressure coefficient C_p is shown in Fig. 7.14 at a relative spanwise coordinate of $\eta = 0.38$ and in Fig. 7.15 at $\eta = 0.78$. The results from both time stepping schemes for the turbulence equations are presented together with experimental results. Except for the shock position at $\eta = 0.38$ the computed pressure distributions meet the experimental data well and do not show obvious differences. At other spanwise coordinates these good agreements between computations and experiments and among the computational results are also found.
The aerodynamic coefficients' convergence at $\mathrm{Re} = 43 \cdot 10^6$ is plotted in Fig. 7.16. While the convergence for all coefficients on the coarsest grid is sped up by the implicit treatment of the turbulence equations, this speed–up is given on the two finer grids in particular for friction drag. Furthermore, the lift coefficient reaches a steady state faster on the finest grid applying the numerical improvements for the turbulence equations compared to their standard treatment. The convergence of the aerodynamic coefficients at $\mathrm{Re} = 86 \cdot 10^6$, Fig. 7.17, shows identical behaviour. A close–up of the convergence of the lift coefficient on the finest grid at these two Reynolds numbers is given in Fig. 7.18. After 500 iterations on the finest grid the lift has reached steady state if the turbulence equations are treated implicitly on the finest grid only. Performing multigrid for the turbulence equations leads to not fully converged lift coefficients at both Reynolds numbers. The lift coefficient is fluctuating over less than one lift count but tends to converge to the lift coefficient based on the DDADI treated k–ω equations. The same behaviour is observed for the coefficient of total drag in Fig. 7.19.
For the 2D calculations at the RAE 2822 airfoil no such speed–up of the convergence of the aerodynamic coefficients has been observed.

Since at $\mathrm{Re} = 129 \cdot 10^6$ no results obtained by multigrid treatment of the turbulence equations exist a preliminary investigation of the effect of `CFLTU` on the convergence of the aerodynamic coefficients is shown in Fig. 7.20. Instead of the default setting of `CFLTU` it has been set to 1000 on all levels. It can be seen that all coefficients experience a speed–up due to this increase of `CFLTU`. This is most obvious in the case of the lift coefficient.

For these 3D calculations the memory required by the FLOWer code has been increased by about 7% using the DDADI scheme instead of the baseline numerical treatment for the turbulence equations. This increase is of the same order as found for the 2D calculations. The CPU efficiency was not found to be increased when the DDADI scheme was applied to the turbulence equations, unlike the behaviour for the 2D calculations.

7.1.3 Conclusions for ALVAST Wing/Body Configuration

The grid generation approach introduced for 2D grids yields Re–adapted grids with well resolved boundary layers for the ALVAST wing/body configuration.

Application of implicit singlegrid treatment of turbulence equations together with minimized coupling results in an improved, i. e. smoother convergence behaviour for all residuals in 3D. Additionally it gives sufficient robustness at Re $= 129 \cdot 10^6$ to converge. In 3D a significantly faster convergence of the aerodynamic coefficients to steady state based on the DDADI scheme for the turbulence equations has been shown in difference to 2D results. Furthermore, the improved numerical treatment of the turbulence equations yields the ability to increase convergence speed of aerodynamic coefficients considerably by increasing the CFL number of the DDADI scheme.

The memory requirements are raised by the same order for 2D and 3D calculations due to the application of the DDADI scheme. However, the CPU time per iteration in 3D has not been changed considerably compared to the baseline FLOWer code.

The improved numerical treatment of the turbulence equations has demonstrated its significantly increased robustness for 3D applications. This is paid off by a slight increase in memory requirement. The increase in CPU time per iteration has to be weighted against the much larger speed–up of the convergence of aerodynamic coefficients. Thus the computational efficiency for 3D applications has been increased.

7.2 ALVAST High Lift Configuration

In this section results for the ALVAST high lift configuration are presented. The grid generation for this configuration was a final milestone for the grid generation package MegaCads [17] at the end of the MEGAFLOW project [66] in 1998. The resulting grid consists of 48 blocks with about 8 million grid points in total. The surface grid (on the second grid level) is shown in Fig. 7.21 together with the pressure distribution.

The numerical simulation of this test case using the DLR flow solver FLOWer was also a final milestone of the MEGAFLOW project. In order to show the improvements introduced in the FLOWer code within the successor project MEGAFLOW II the same test case has been calculated once more at the end of this project in 2002. All computations have been performed by J. Raddatz [93]. The flow conditions of this test case are $M_\infty = 0.22$ and $\alpha = 12.03^\circ$ and Re $= 2 \cdot 10^6$.

Fig. 7.22 shows the convergence of the density residual for both calculations. The computation labeled "CC : RANS – 2G ; k–ω – DDADI + Δt_{tu}" has been performed using the cell centered metric and a different type of coefficients for the implicit residual smoothing [120] beside the DDADI scheme for turbulence equations. The latter enabled the use of two multigrid levels for the RANS equations — the grid topology did not allow for more levels — while the "CV : RANS – SG ; k–ω – RK–SG + Δt_{me}" labeled computation was a pure singlegrid calculation based on the cell vertex metric. Additionally the corrected time step calculation for the turbulence equations has been used for the DDADI based results. Furthermore the turn–around–time has been reduced for the new computation by parallelization. With the current FLOWer code the calculation has been performed within about 10 hours, i. e. over night on four processors of a NEC SX5 vector computer. About nine days pure CPU time would have been needed with the old FLOWer code ("CV : RANS – SG ; k–ω – RK–SG + Δt_{me}") for this calculation on the same computer.

Although the density residual based on DDADI treatment of the turbulence equations in Fig. 7.22 is less converged than the density residual based on the original treatment of the k–ω equations, the lift coefficient — shown in Fig. 7.23 — is already converged to a comparable level. The turbulence residuals are shown in Fig. 7.24 and Fig. 7.25, respectively. In particular the residual of the ω equation shows a smoother convergence for the DDADI computation. Both residuals show a shift between the computations. This shift is due to a change in the calculation of the convergence data. At the end of the MEGAFLOW project the convergence data for the turbulence equations was based on the point implicitly modified time step Δt_{pi} as given in Eq. (3.62) instead of the time step Δt_{tu}.

These computations show two things. Firstly the improved numerical treatment of the turbulence equations can be used in consideration of nearly any other feature of the FLOWer code. Secondly the robustness and efficiency of the computation of complex 3D configurations has been increased considerably by combining the results of this work with other new features of the FLOWer code, especially the cell centered metric.

7.3 OA 415 Rotor Blade Airfoil (2D)

This test case is based on a rotor blade profile. It has been provided by Eurocopter. The CAD repair of the tab with its open trailing edge made it necessary to investigate the effects of different strategies to generate a closed profile on the numerical simulation. The two approaches taken are briefly described in Sect. 7.3.1. The numerical calculations are discussed afterwards in Sect. 7.3.2. In Sect. 7.3.3 conclusions are given.

7.3.1 Geometry Modifications for OA 415 Rotor Blade Airfoil and Grid Generation

The OA 415 airfoil — as shown in the upper half of Fig. 7.26 — is a rotor blade profile. Due to the industrial relevance of this airfoil the geometrical information of the plots has been fudged. Its trailing edge runs into a tab. The geometry data provided do not contain any information about the shape of the trailing edge. In manufacturing this information is not as necessary as the trailing edge has a certain thickness due to material constraints. A straight line between the upper and the lower end of the trailing edge may be assumed.
For the numerical simulation the treatment of the trailing edge of any airfoil is a crucial point. Assuming a straight line also for the numerical computation requires an extremely fine resolution of the trailing edge area in order to capture all physical effects. Furthermore, a special grid block needs to be introduced for the resolution of a thick trailing edge.

Two other possibilities of generating a closed trailing edge have been looked at. Both approaches result in simple one block C-grids.
On the one hand the upper and lower part of the profile are rotated around the nose point until the trailing edge is closed. This results in a very sharp trailing edge. The thickness of the geometry is slightly changed by this approach. The nose area is also

modified.
The second approach is to deform only the outermost upper and lower airfoil parts gaining a blunt trailing edge. Both trailing edges are shown in the lower half of Fig. 7.26. The corresponding effects on the aerodynamic characteristics of the airfoil are not discussed here.

Grids suitable for Navier–Stokes calculations for both geometry versions have been generated by adding a couple of boundary layer resolving grid lines to Euler grids given. This is a standard approach to ease grid generation and to achieve comparability of grid quality between Euler and Navier–Stokes grids.

7.3.2 Computational Results for Different Geometries of OA 415 Rotor Blade Airfoil

Two computations have been performed for each geometry variant: One with multigrid treatment of the k–ω equations in use and the other with implicit singlegrid treatment of these equations.

Figs. 7.27 – 7.30 show the convergence behaviour of all residuals and of the lift coefficient for the sharp trailing edge. The RANS equations are always treated by a three level W–cycle. For the turbulence equations on the one hand the same treatment as for the RANS equations has been applied. On the other hand they are treated only singlegrid but with the DDADI scheme and a correspondingly higher CFL number. Successive grid refinement applied to all equations. At first 100 iterations on the third grid level are performed followed by 300 iterations on the second grid level. On the finest grid level 2000 multigrid cycles are iterated.
The higher robustness due to the implicit singlegrid treatment of the turbulence equations can be easily seen from Figs. 7.27 – 7.29. The convergence becomes smoother and more efficient compared to the original FLOWer approach. Again this holds not only for the turbulence residuals but also for the density residual.
After 1800 iterations the oscillations in all residual convergence plots vanish for the original FLOWer approach likewise. At the same time the convergence of the lift coefficient — Fig. 7.30 — starts with some smooth, but new oscillations. The implicit treatment of the turbulence equations already provides a steady lift coefficient after 1000 iterations.

The results obtained for the blunt trailing edge case are documented in Figs. 7.31 – 7.34. Although the multigrid treatment of the whole system of equations has been reduced to a two level cycle the calculation diverges. This divergence is represented in the lift coefficient's convergence — Fig. 7.34 — by increasing oscillations. The discontinuity of the geometry of the blunt trailing edge results in larger velocity gradients in its vicinity. These gradients enlarge the production of turbulence which may lead — amplified by multigrid — to the destabilization of this computation.
Switching to the implicit treatment of the turbulence equations using DDADI within a singlegrid approach enables the use of a 4 level W–cycle for the RANS equations. All residuals, Figs. 7.31 – 7.33, show a smooth convergence behaviour. The same holds for the lift coefficient in Fig. 7.34 which is converged within 1000 iterations.

7.3.3 Conclusions for OA 415 Rotor Blade Airfoil

By this two dimensional test case the effect of the numerical treatment of turbulence equations in case of geometry induced convergence problems has been investigated. It has been clearly demonstrated that the robustness increase gained by the improved numerical treatment of the turbulence equations is able to overcome such convergence problems, too. Additionally it has been shown that the convergence of aerodynamic coefficients may be sped up significantly even for 2D test cases.

8 Conclusions

This work aimed at providing numerical methods that enable the robust and efficient simulation of turbulent flows around civil transport aircraft configurations at flight Reynolds numbers. The combined utilization of a computational and a theoretical approach for the systematic investigation of the numerical treatment of turbulence equations characterizes the work at hand.

This work is based on the DLR CFD tool FLOWer. Its numerical treatment of transport equation turbulence models has been presented in operator notation. For these operators corresponding Fourier symbols have been derived. This led to a consistent notation for the numerical treatment of turbulence equations and its Fourier analysis. The FLOWer code has been employed as a computational tool while the Fourier analysis serves as a theoretical approach for the systematic investigation of the numerical treatment of turbulence equations.

For computational investigations at several high or flight Reynolds numbers the RAE 2822 airfoil has been chosen as a two dimensional test case. Computational grids have been equipped with a Reynolds number adapted boundary layer resolution.
While for the algebraic turbulence model of Baldwin and Lomax [4] even at flight Reynolds numbers no robustness problems were encountered, the multigrid treatment of transport equation turbulence models, here the k–ω turbulence model according to Wilcox [131], has been found to deteriorate robustness. These conclusions based on computational results have been corroborated by results from Fourier analysis. By means of this Fourier analysis the influence of (turbulence) source terms in particular on the numerical properties of a multigrid cycle have been investigated. The observed deterioration is based on a destabilizing effect of productive (turbulence) source terms which is amplified in the framework of multigrid. In addition destructive source terms were found to reduce the efficiency of multigrid methods.
In general the obtained results hold for any transport equation turbulence model since the Fourier analysis is independent of the turbulence equations actually employed for the computational investigations.

Modifications to the prolongation operator for the coarse grid corrections of the turbulence equations have been investigated. All robustness improvements obtained were based on a reduction of the influence of the coarser grids on the fine grid solution of the turbulence equations. A pure singlegrid treatment of turbulence equations with an unchanged multigrid treatment of the RANS equations showed highest robustness improvements up to flight Reynolds numbers compared to other multigrid modifications investigated.
The convergence of the RANS equations, represented by the density residual, has been

improved by reducing the coupling of the k–ω turbulence model to the RANS equations to the eddy viscosity only. However, this minimized coupling is physically incorrect for this kind of turbulence model since it provides, besides the eddy viscosity, for instance the turbulent kinetic energy k which has to be considered within the RANS equations for a physically correct coupling.
The convergence of the turbulence equations — treated only on the finest grid — has been improved by a separate time step which is based on the eigenvalues of the turbulence equations. An alternative linearization of the point implicitly treated source term did not show any improvements to the robustness or the efficiency of the numerical method in computational applications.

The breakthrough concerning robustness and efficiency has been achieved by a line or fully implicit treatment of the turbulence equations. The j–line implicit time integration of the singlegrid treated turbulence equations enabled convergence to machine accuracy also for the highest Reynolds number investigated for the RAE 2822 airfoil, $\mathrm{Re} = 100 \cdot 10^6$. The convergence of all residuals showed a reduced Reynolds number influence compared to corresponding computations using the algebraic turbulence model of Baldwin and Lomax.
The fully implicit treatment of the turbulence equations is based on a diagonally dominant variant of the ADI method named DDADI. Its full potential can be exploited for three dimensional applications in particular. The ALVAST wing/body configuration has been chosen as a three dimensional test case of industrial relevance. As for the RAE 2822 airfoil, computational grids with a Reynolds number adapted boundary layer resolution have been generated for this configuration. The application of the DDADI scheme to this test case demonstrated an essentially improved robustness for the numerical simulation of turbulent flows around civil transport aircraft at flight Reynolds numbers.

A Annex

A.1 Central Discretization of Inviscid Terms

The inviscid terms of the Reynolds averaged Navier–Stokes equations, Eq. (2.6), are subject to a spatially central discretization scheme which is of second order accuracy on regular grids. Here this discretization is presented based on the scalar example equation introduced in Sect. 2.4.

The surface integral of the inviscid flux, Eq. (2.54), is taken over the supercell given in Eq. (3.2) which yields

$$\begin{aligned}\int\limits_{\partial\widetilde{V}_{i,j}} -\bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}(\partial\widetilde{V}_{i,j}) \;=\; & \int\limits_{\partial V_{I,J}} -\bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}(\partial V_{I,J}) + \int\limits_{\partial V_{I-1,J}} -\bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}(\partial V_{I-1,J}) \\ & + \int\limits_{\partial V_{I,J-1}} -\bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}(\partial V_{I,J-1}) + \int\limits_{\partial V_{I-1,J-1}} -\bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}(\partial V_{I-1,J-1})\,.\end{aligned} \tag{A.1}$$

Since the supercell approach yields a set of overlapping control volumes it has to be assured that the overlaps do not distort the solution in the computational domain. In 2D each grid cell contributes to four supercells while in 3D eight supercells take contributions from the same grid cell. In order to recover from these overlaps only one fourth (respectively one eighth in 3D) of the upper right hand side is taken:

$$\begin{aligned}\int\limits_{\partial\widetilde{V}_{i,j}} -\bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}(\partial\widetilde{V}_{i,j}) \;=\; & \frac{1}{4}\left(\int\limits_{\partial V_{I,J}} -\bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}(\partial V_{I,J}) + \int\limits_{\partial V_{I-1,J}} -\bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}(\partial V_{I-1,J}) \right. \\ & \left. + \int\limits_{\partial V_{I,J-1}} -\bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}(\partial V_{I,J-1}) + \int\limits_{\partial V_{I-1,J-1}} -\bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}(\partial V_{I-1,J-1}) \right)\,.\end{aligned} \tag{A.2}$$

Each of these integrals is split into four integrals over the cell faces of the corresponding control volume, as shown below for $V_{I,J}$:

$$\begin{aligned}\int\limits_{\partial V_{I,J}} -\bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}(\partial V_{I,J}) \;=\; & \int\limits_{C_{I+\frac{1}{2},J}} \bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}C_{I+\frac{1}{2},J} - \int\limits_{C_{I-\frac{1}{2},J}} \bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}C_{I-\frac{1}{2},J} \\ & - \int\limits_{C_{I,J+\frac{1}{2}}} \bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}C_{I,J+\frac{1}{2}} + \int\limits_{C_{I,J-\frac{1}{2}}} \bar{\bar{F}}_c \bullet \vec{n}\,\mathrm{d}C_{I,J-\frac{1}{2}}\,.\end{aligned} \tag{A.3}$$

Here it is assumed that the cell face normal unit vector $\vec{n}$ on each cell face is oriented positive to the corresponding curvilinear coordinate direction (Figs. 3.4 and 3.2).
By applying the mean value theorem of integral calculus to each of the surface integrals in Eq. (A.3), it follows

$$\begin{aligned} \int\limits_{C_{I+\frac{1}{2},J}} \overline{\overline{F}}_c \bullet \vec{n}\, \mathrm{d}C_{I+\frac{1}{2},J} &= \overline{\overline{F}}_{c;\, I+\frac{1}{2},J} \bullet \vec{n}_{I+\frac{1}{2},J} \int\limits_{C_{I+\frac{1}{2},J}} \mathrm{d}C_{I+\frac{1}{2},J} \\ &= \overline{\overline{F}}_{c;\, I+\frac{1}{2},J} \bullet \vec{n}_{I+\frac{1}{2},J}\, \mathcal{C}_{I+\frac{1}{2},J}\,. \end{aligned} \tag{A.4}$$

$\vec{n}_{I+\frac{1}{2},J}$ denotes the cell face normal unit vector of the cell face $C_{I+\frac{1}{2},J}$ as given in Fig. 3.4. Using $_{I+\frac{1}{2},J}$ as an index for $\overline{\overline{F}}_c$ indicates that this value is taken on the cell face $C_{I+\frac{1}{2},J}$. As the outer normal unit vector $\vec{n}_{I+\frac{1}{2},J}$ and the cell face area $\mathcal{C}_{I+\frac{1}{2},J}$ are known only the integral mean value of the flux tensor $\overline{\overline{F}}_c$ on the cell face $C_{I+\frac{1}{2},J}$ has to be approximated. As the values of W are located in the cell vertices $(i+1,j)$ and $(i+1,j+1)$ the value of $\overline{\overline{F}}_c$ on $C_{I+\frac{1}{2},J}$ is calculated by means of the average of W on the same cell face $C_{I+\frac{1}{2},J}$:

$$\overline{\overline{F}}_{c;\, I+\frac{1}{2},J} \approx \overline{\overline{F}}_c\left(W_{I+\frac{1}{2},J}\right) \qquad \text{with} \quad W_{I+\frac{1}{2},J} = \frac{1}{2}\left(W_{i+1,j} + W_{i+1,j+1}\right)\,. \tag{A.5}$$

Inserting this formula into Eq. (A.4) and using this in Eq. (A.3) results in a central difference scheme of second order assuming a cartesian grid with constant grid sizes. Even in case of discontinuities in body fitted grids the scheme degrades to first order only [102].

Because the central discretization of the inviscid terms does not damp high error frequencies artificial dissipation has to be used. Suitable terms have been introduced by Jameson et al. [56], while a more detailed explanation is given in [54]. As the artificial dissipation has been discussed in extent e. g. by Kroll et al. [63] and no artificial dissipation is needed for the upwind discretization of the turbulence equations, the reader is referred to the literature.

Comparing Eq. (A.3) with Eq. (3.7) and Eq. (A.4) with Eq. (3.8) respectively shows that the basic difference between the central and the upwind discretization is in the approximation of the flux tensor $\overline{\overline{F}}_c$ on the corresponding cell face $C_{I+\frac{1}{2},J}$ resp. $C_{i+\frac{1}{2},j}$. Additionally the supercell approach results in a nine point stencil for the discretization of the inviscid terms in two dimensions. On the other hand a five point stencil would be obtained for the central discretization based on the compact cell.

A.2 Approximation of Viscous Terms

The derivatives of W with respect to x and y in the point $(i, j+\frac{1}{2})$ are computed using a local transformation from cartesian coordinates (x, y) to generalized curvilinear coordinates (ξ, η) as given by Radespiel et al. [97]. This coordinate transformation results in

$$\left(\frac{\partial W}{\partial x}, \frac{\partial W}{\partial y}\right) = \left(\frac{\partial W}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial W}{\partial \eta}\frac{\partial \eta}{\partial x}\,,\; \frac{\partial W}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial W}{\partial \eta}\frac{\partial \eta}{\partial y}\right)\,. \tag{A.6}$$

This is simplified according to the thin shear layer approximation, i. e. only gradients in the direction normal to the viscous shear layers are taken into account. Since η is usually assumed to be the wall normal coordinate direction of the curvilinear coordinate system it is taken to be perpendicular to the viscous shear layers in general.* Thus Eq. (A.6) is approximated by

$$\left(\frac{\partial W}{\partial x}, \frac{\partial W}{\partial y}\right) \approx \left(\frac{\partial W}{\partial \eta}\frac{\partial \eta}{\partial x}, \frac{\partial W}{\partial \eta}\frac{\partial \eta}{\partial y}\right). \tag{A.7}$$

Some vector algebra as explained in extent by Anderson [1] as well as by Borisenko and Tarapov [8] yields

$$\left(\frac{\partial W}{\partial \eta}\frac{\partial \eta}{\partial x}, \frac{\partial W}{\partial \eta}\frac{\partial \eta}{\partial y}\right) = \left(\frac{\partial W}{\partial \eta}\frac{1}{|J|}\left[-\frac{\partial y}{\partial \xi}\right], \frac{\partial W}{\partial \eta}\frac{1}{|J|}\frac{\partial x}{\partial \xi}\right) = \frac{\partial W}{\partial \eta}\frac{1}{|J|}\left(-\frac{\partial y}{\partial \xi}, \frac{\partial x}{\partial \xi}\right) \tag{A.8}$$

with the determinant of the Jacobian of the transformation from the cartesian coordinates (x, y) to the curvilinear coordinates (ξ, η) defined as

$$|J| = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}. \tag{A.9}$$

Furthermore, the Jacobian determinant $|J|$ appearing in Eq. (A.8) equals the volume of a suitable control cell surrounding grid point $(i, j+\frac{1}{2})$

$$\left|J_{i,j+\frac{1}{2}}\right| = \left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}\right)_{i,j+\frac{1}{2}} = \widetilde{\mathcal{V}}_{i,j+\frac{1}{2}}. \tag{A.10}$$

From some vector algebra as presented by Borisenko and Tarapov [8] or Rossow et al. [103] it can be easily derived that $(-\partial y/\partial \xi, \partial x/\partial \xi)$ is a contravariant vector to the curvilinear coordinate system. Since a contravariant vector can be calculated from a cross product of covariant vectors it equals a suitable cell face area times the corresponding cell face normal unit vector.
From Fig. 3.1 and Fig. 3.4 respectively it can be seen that the cell faces from $(i-1/2, j+1)$ to $(i+1/2, j+1)$ and from $(i-1/2, j)$ to $(i+1/2, j)$ together with the volume $\mathcal{V}_{i,j+\frac{1}{2}}$ of the auxiliary cell as defined in Eq. (3.35) are suitable for the approximation in $(i, j+\frac{1}{2})$. These cell faces are approximated by

$$\frac{1}{2}\left[(\mathcal{C}\vec{n})_{I-1,J+\frac{1}{2}} + (\mathcal{C}\vec{n})_{I,J+\frac{1}{2}}\right] \qquad \text{and} \qquad \frac{1}{2}\left[(\mathcal{C}\vec{n})_{I-1,J-\frac{1}{2}} + (\mathcal{C}\vec{n})_{I,J-\frac{1}{2}}\right] \tag{A.11}$$

respectively. This selection yields

$$\begin{aligned}&\left[\frac{\partial W}{\partial \eta}\frac{1}{|J|}\left(-\frac{\partial y}{\partial \xi}, \frac{\partial x}{\partial \xi}\right)\right]_{i,j+\frac{1}{2}} \\ &\approx \frac{W_{i,j+1}\dfrac{\frac{1}{2}\left[(\mathcal{C}\vec{n})_{I-1,J+\frac{1}{2}} + (\mathcal{C}\vec{n})_{I,J+\frac{1}{2}}\right]}{\mathcal{V}_{i,j+\frac{1}{2}}} - W_{i,j}\dfrac{\frac{1}{2}\left[(\mathcal{C}\vec{n})_{I-1,J-\frac{1}{2}} + (\mathcal{C}\vec{n})_{I,J-\frac{1}{2}}\right]}{\mathcal{V}_{i,j+\frac{1}{2}}}}{\eta_{i,j+1} - \eta_{i,j}}.\end{aligned} \tag{A.12}$$

*Within the FLOWer code a "full thin shear layer approximation" has been implemented: The user has to define which coordinate directions are to be treated as perpendicular to visous shear layer.

Since $\eta_{i,j+1} - \eta_{i,j} = 1$ one gets

$$\left[\frac{\partial W}{\partial \eta}\frac{1}{|J|}\left(-\frac{\partial y}{\partial \xi}, \frac{\partial x}{\partial \xi}\right)\right]_{i,j+\frac{1}{2}} \approx \frac{W_{i,j+1}\frac{1}{2}\left[(\mathcal{C}\vec{n})_{I-1,J+\frac{1}{2}} + (\mathcal{C}\vec{n})_{I,J+\frac{1}{2}}\right] - W_{i,j}\frac{1}{2}\left[(\mathcal{C}\vec{n})_{I-1,J-\frac{1}{2}} + (\mathcal{C}\vec{n})_{I,J-\frac{1}{2}}\right]}{\mathcal{V}_{i,j+\frac{1}{2}}}. \tag{A.13}$$

This formula exhibits a problem of a finite volume approach if — due to damping properties — as few points as possible for the discretization stencil (at minimum like here 2 points) are to be used: Freestream consistency is defined as zero gradients of a quantity if a constant distribution of this quantity is given. Thus freestream consistency is necessary for any spatial discretization of derivatives.
Assuming $W_{i,j+1}$ and $W_{i,j}$ to be equal (constant distribution of W) freestream consistency of Eq. (A.13) is given if and only if

$$(\mathcal{C}\vec{n})_{I-1,J+\frac{1}{2}} + (\mathcal{C}\vec{n})_{I,J+\frac{1}{2}} = (\mathcal{C}\vec{n})_{I-1,J-\frac{1}{2}} + (\mathcal{C}\vec{n})_{I,J-\frac{1}{2}} \tag{A.14}$$

is fulfilled. Since this equality is highly mesh dependent it is not given in general.
In order to overcome this deficiency Eq. (A.8) is not approximated via a finite volume approach but a finite difference approach, i. e. instead of approximating the product of derivative, reciprocal of Jacobian determinant and contravariant vector as one term each factor is approximated on its own: The Jacobian determinant is still given by Eq. (A.10). The derivative of W is approximated by

$$\left(\frac{\partial W}{\partial \eta}\right)_{i,j+\frac{1}{2}} \approx \frac{W_{i,j+1} - W_{i,j}}{\eta_{i,j+1} - \eta_{i,j}} = W_{i,j+1} - W_{i,j} \tag{A.15}$$

while the contravariant vector is determined as the directed area of the average over those cell faces used for the finite volume approach:

$$\left(-\frac{\partial y}{\partial \xi}, \frac{\partial x}{\partial \xi}\right)_{i,j+\frac{1}{2}} \approx \frac{1}{4}\left[(\mathcal{C}\vec{n})_{I-1,J+\frac{1}{2}} + (\mathcal{C}\vec{n})_{I,J+\frac{1}{2}} + (\mathcal{C}\vec{n})_{I-1,J-\frac{1}{2}} + (\mathcal{C}\vec{n})_{I,J-\frac{1}{2}}\right]. \tag{A.16}$$

This determination of the contravariant vector corresponds to forcing Eq. (A.14).
Collecting these three approximations yields

$$\left(\frac{\partial W}{\partial \eta}\frac{1}{|J|}\left(-\frac{\partial y}{\partial \xi}, \frac{\partial x}{\partial \xi}\right)\right)_{i,j+\frac{1}{2}} \approx (W_{i,j+1} - W_{i,j})\frac{\frac{1}{4}\sum_{l=I-1}^{I}\left[(\mathcal{C}\vec{n})_{l,J+\frac{1}{2}} + (\mathcal{C}\vec{n})_{l,J-\frac{1}{2}}\right]}{\mathcal{V}_{i,j+\frac{1}{2}}}. \tag{3.34}$$

This finite difference based formula ensures a freestream consistent discretization of the viscous fluxes.

Remark A.1 *Another point of view is to say that the partial derivatives $\partial\eta/\partial x$ and $\partial\eta/\partial y$ in Eq. (A.7) (sometimes called "metric derivatives" or "direct metrics" or even "metrics") are approximated by their geometrical representation:*

$$\left(\frac{\partial \eta}{\partial x}, \frac{\partial \eta}{\partial y}\right)_{i,j+\frac{1}{2}} \approx \frac{1}{\mathcal{V}_{i,j+\frac{1}{2}}} \cdot \frac{(\mathcal{C}\vec{n})_{I-1,J+\frac{1}{2}} + (\mathcal{C}\vec{n})_{I,J+\frac{1}{2}} + (\mathcal{C}\vec{n})_{I-1,J-\frac{1}{2}} + (\mathcal{C}\vec{n})_{I,J-\frac{1}{2}}}{4}. \tag{A.17}$$

The result is the same as derived above.

A.3 Operator Notation for Runge–Kutta Scheme

In order to get an operator notation of the standard 5-stage Runge–Kutta scheme

$$
\begin{aligned}
W^{(0)} &= W^n \\
W^{(1)} &= W^{(0)} + \alpha_1 \tau R^{(0)} \\
W^{(2)} &= W^{(0)} + \alpha_2 \tau R^{(1)} \\
W^{(3)} &= W^{(0)} + \alpha_3 \tau R^{(2)} \\
W^{(4)} &= W^{(0)} + \alpha_4 \tau R^{(3)} \\
W^{(5)} &= W^{(0)} + \alpha_5 \tau R^{(4)} \\
W^{n+1} &= W^{(5)}
\end{aligned}
\tag{3.54}
$$

it is necessary to rewrite the formulae as a product of W^n. $R^{(q-1)}$ is given by Eq. (3.56). As the convective term is evaluated on each stage a successive inserting has to be done. Extending the second stage according to Eq. (3.56) and inserting the first stage into the right hand side of the second stage yields

$$
\begin{aligned}
W^{(2)} &= W^{(0)} + \alpha_2 \tau \left[Z_c\circ\, W^{(1)} + (Z_v\circ\, + Z_S\circ\,)\, W^{(0)} \right] \\
&= W^{(0)} + \alpha_2 \tau \left[Z_c\circ\, \left(W^{(0)} + \alpha_1 \tau R^{(0)} \right) + (Z_v\circ\, + Z_S\circ\,)\, W^{(0)} \right] .
\end{aligned}
$$

Elimination of the most outer brackets results in

$$
\begin{aligned}
W^{(2)} &= W^{(0)} + \alpha_2 \tau \left[Z_c\circ\, \left(W^{(0)} + \alpha_1 \tau R^{(0)} \right) + (Z_v\circ\, + Z_S\circ\,)\, W^{(0)} \right] \\
&= W^{(0)} + \alpha_2 \tau Z_c\circ\, W^{(0)} + \alpha_2 \tau Z_c\circ\, \alpha_1 \tau R^{(0)} + \alpha_2 \tau\, (Z_v\circ\, + Z_S\circ\,)\, W^{(0)} .
\end{aligned}
$$

By rearranging and collecting the summands one achieves a formula for the second stage which does no longer depend on values different from the starting values:

$$
\begin{aligned}
W^{(2)} &= W^{(0)} + \alpha_2 \tau Z_c\circ\, W^{(0)} + \alpha_2 \tau Z_c\circ\, \alpha_1 \tau R^{(0)} + \alpha_2 \tau\, (Z_v\circ\, + Z_S\circ\,)\, W^{(0)} \\
&= W^{(0)} + \alpha_2 \tau R^{(0)} + \alpha_2 \alpha_1 \tau Z_c\circ\, \tau R^{(0)} \\
&= W^{(0)} + (\alpha_2 + \alpha_2 \alpha_1 \tau Z_c\circ\,)\, \tau R^{(0)} .
\end{aligned}
\tag{A.18}
$$

This equation holds only for the case that neither the viscous terms nor the source term is evaluated on the second stage. Since this is the default for the application of the FLOWer code a more general Runge–Kutta method is not derived here.
Inserting Eq. (A.18) to the detailed third stage of the standard Runge–Kutta scheme Eq. (3.54) gives

$$
\begin{aligned}
W^{(3)} &= W^{(0)} + \alpha_3 \tau \left[Z_c\circ\, W^{(2)} + (Z_v\circ\, + Z_S\circ\,)\, W^{(0)} \right] \\
&= W^{(0)} + \left(\alpha_3 + \alpha_3 \alpha_2 \tau Z_c\circ\, + \alpha_3 \alpha_2 \alpha_1 \left(\tau Z_c\circ\, \right)^2 \right) \tau R^{(0)} .
\end{aligned}
\tag{A.19}
$$

This formulation of the third stage also only depends on start values. Generalizing Eq. (A.18) and Eq. (A.19) to an arbitrary number of Runge–Kutta stages yields

$$
W^{(q)} = W^{(0)} + \left[\alpha_q + \alpha_q \alpha_{q-1} \tau Z_c\circ\, + \ldots + \prod_{r=1}^{q} \alpha_r \left(\tau Z_c\circ\, \right)^{q-1} \right] \tau R^{(0)} .
\tag{A.20}
$$

The terms inside the square brackets equal a characterizing operator polynomial for the qth stage of the Runge–Kutta scheme

$$f_{\mathrm{ex}}^{(q)} \circ \; = \alpha_q + \alpha_q \alpha_{q-1} \tau Z_c \circ \; + \ldots + \prod_{r=1}^{q} \alpha_r \left(\tau Z_c \circ \right)^{q-1} . \tag{3.57}$$

A.3.1 Operator Notation for Runge–Kutta Scheme including Point Implicit Treatment of Source Terms

The same procedure yields a characterizing operator polynomial for the qth stage of the Runge–Kutta scheme with a point implicit treatment of the source term. The only difference lies in the stage coefficient which is modified by the implicit contribution of the source term as derived in Sect. 3.2.2.2. This can be seen by comparing the qth stage for the purely explicit scheme

$$W^{(q)} = W^{(0)} + \alpha_q \tau R^{(q-1)} \tag{3.55}$$

to the qth stage of the Runge–Kutta scheme with point implicit treatment of the source term

$$W^{(q)} = W^{(0)} + \alpha_q^{\mathrm{pi-src}} \tau R^{(q-1)} . \tag{3.66}$$

$R^{(q-1)}$ has been defined in Eq. (3.56) and $\alpha_q^{\mathrm{pi-src}}$ is given by Eq. (3.67). The resulting characterizing operator polynomial including the point implicitly treated source term reads

$$f_{\mathrm{pi-src}}^{(q)} \circ \; = \alpha_q^{\mathrm{pi-src}} + \alpha_q^{\mathrm{pi-src}} \alpha_{q-1}^{\mathrm{pi-src}} \tau Z_c \circ \; + \ldots + \prod_{r=1}^{q} \alpha_r^{\mathrm{pi-src}} \left(\tau Z_c \circ \right)^{q-1} . \tag{3.68}$$

A.3.2 Operator Notation for Runge–Kutta Scheme including Implicit Residual Smoothing

If implicit residual smoothing is applied the qth stage reads

$$W^{(q)} = W^{(0)} + B^{-1} \circ \alpha_q \tau R^{(q-1)} \tag{A.21}$$

with $R^{(q-1)}$ given by Eq. (3.56). This stage residual does not include implicit residual smoothing on previous stages. If $B^{-1}\circ$ is applied on every stage successive inserting as done above yields

$$\begin{aligned} W^{(q)} \;\; = \;\; & W^{(0)} + B^{-1} \circ \Bigg[\alpha_q + \alpha_q \alpha_{q-1} \tau Z_c \circ B^{-1} \circ \\ & + \ldots + \prod_{r=1}^{q} \alpha_r \left(\tau Z_c \circ B^{-1} \circ \right)^{q-1} \Bigg] \tau R^{(0)} . \end{aligned} \tag{A.22}$$

The characterizing operator polynomial for the qth stage of the Runge–Kutta scheme equals $B^{-1}\circ$ times the terms enclosed in square brackets including implicit residual smoothing on every stage. In combination with the point implicit treatment of the source term the polynomial is yielded as

$$
\begin{aligned}
f^{(q)}_{\mathrm{pi-src,\,IRS}}\circ \;&=\; B^{-1}\circ\Bigg[\,\alpha_q^{\mathrm{pi-src}}+\alpha_q^{\mathrm{pi-src}}\alpha_{q-1}^{\mathrm{pi-src}}\tau Z_c\circ B^{-1}\circ \\
&\qquad +\ldots+\prod_{r=1}^{q}\alpha_r^{\mathrm{pi-src}}\left(\tau Z_c\circ B^{-1}\circ\right)^{q-1}\Bigg]\,.
\end{aligned}
\tag{3.83}
$$

A.3.3 Operator Notation for Runge–Kutta Scheme on Coarse Grids

Also for the coarse grid operator introduced in Sect. 3.3.3.3 in general the same procedure holds. For the derivation of the coarse grid operator an "additional" step has to be performed: When using the modified stage residual $W_{2h}^{(0)}$ (Eq. (3.99)) in Eq. (3.54) it has to be replaced by its definition:

$$
\begin{aligned}
W_{2h}^{(q)} &= W_{2h}^{(0)}+\alpha_q\tau_{2h}\widetilde{R}_{2h}^{(q-1)} \\
&= W_{2h}^{(0)}+\alpha_q\tau_{2h}R_{2h}^{(q-1)}+\alpha_q\tau_{2h}\left(I_h^{2h}\circ R_h^{n+1}-R_{2h}^{(0)}\right)\,.
\end{aligned}
\tag{3.100}
$$

In this equation the original qth stage residual $R_{2h}^{(q-1)}$ as defined in Eq. (3.56) can be inserted. After this the same derivation as above takes place.

A.4 Extension of Implicit Time Stepping Schemes to Time Accurate Computations using Dual Time Stepping

Integrating the basic system $\partial W/\partial t = \overline{\overline{F}}(W)$ in time using an implicit second order accurate backward difference formula yields

$$
\frac{3}{2\Delta t}W^{n+1}-\frac{2}{\Delta t}W^{n}+\frac{1}{2\Delta t}W^{n-1}=Z\circ W^{n+1}\,. \tag{A.23}
$$

This equation can be interpreted as a modified or pseudo steady state equation:

$$
\frac{\partial W^{n+1}}{\partial\tau}=Z_\tau\circ W^{n+1}=Z\circ W^{n+1}-S(W^{n+1}) \tag{A.24}
$$

forming a source term from terms depending on the physical time step Δt,

$$
S(W^{n+1})=\frac{3}{2\Delta t}W^{n+1}-\frac{2}{\Delta t}W^{n}+\frac{1}{2\Delta t}W^{n-1}\,. \tag{A.25}
$$

Here $^{n+1}$ is no longer the iteration index but denotes the pseudo steady state solution to be reached. This system can be solved by an implicit pseudo time stepping scheme as

$$\frac{\Delta_\nu W^{n+1}}{\Delta\tau} = Z \circ W^{n+1;\,\nu+1} - S(W^{n+1;\,\nu+1}) \tag{A.26}$$

using $\Delta_\nu W^{n+1} \stackrel{\text{def}}{=} W^{n+1;\,\nu+1} - W^{n+1;\,\nu}$. Linearization of $\frac{3}{2\Delta t} W^{n+1;\,\nu+1}$ in $S(W^{n+1;\,\nu+1})$ gives

$$\begin{aligned} \frac{3}{2\Delta t} W^{n+1;\,\nu+1} &= \frac{3}{2\Delta t} W^{n+1;\,\nu} + \frac{3}{2\Delta t} \Delta_\nu W^{n+1} \\ \Rightarrow \qquad S(W^{n+1;\,\nu+1}) &= S(W^{n+1;\,\nu}) + \frac{3}{2\Delta t} \Delta_\nu W^{n+1} \end{aligned} \tag{A.27}$$

by which one reaches

$$\left[\frac{1}{\Delta\tau} + \frac{3}{2\Delta t}\right] \Delta_\nu W^{n+1} = Z \circ W^{n+1;\,\nu+1} - S(W^{n+1;\,\nu}) \,. \tag{A.28}$$

The linearization of $Z \circ W^{n+1;\,\nu+1}$ reads — in equivalence to Eq. (6.20) and Eq. (6.22) —

$$Z \circ W^{n+1;\,\nu+1} \approx Z \circ W^{n+1;\,\nu} + \left[Z_c \circ \; + Z_v \circ \; + \frac{\partial (Z_S \circ W^{n+1;\,\nu})}{\partial W} \right] \Delta_\nu W^{n+1} \,. \tag{A.29}$$

From this

$$\left[\mathsf{M} + \frac{3}{2\Delta t} \cdot \mathsf{I} \right] \Delta_\nu W^{n+1} = Z \circ W^{n+1;\,\nu} - S(W^{n+1;\,\nu}) \tag{A.30}$$

is gained in combination with Eq. (6.58) and the corresponding definitions. This equation shows that the extension of any implicit scheme to the dual time stepping approach is very easy.
The same theoretical result has also been presented by Dubuc et al. [31].

B Figures

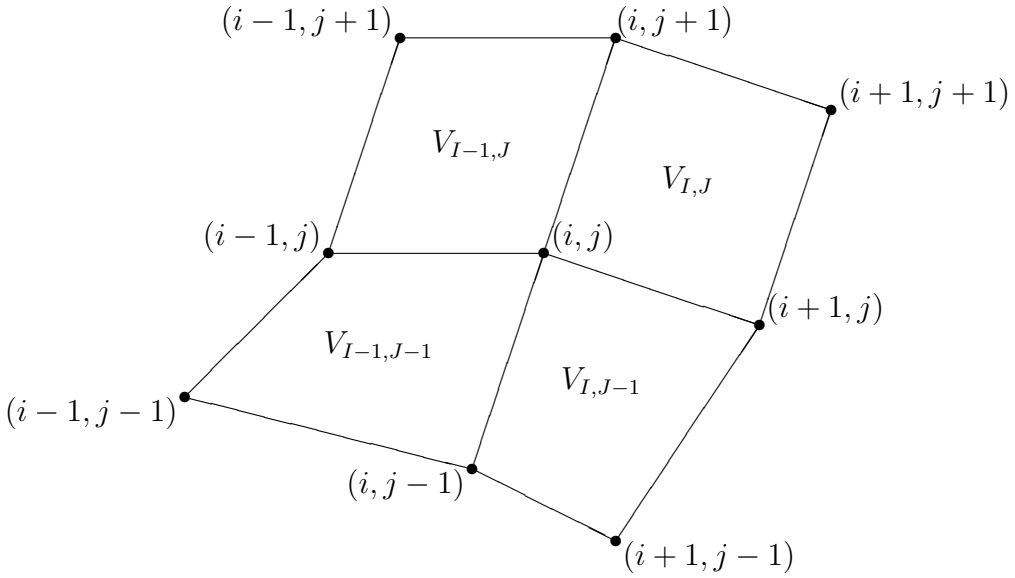

Figure 3.1: Volume and nodal point indices of 2D grid cells

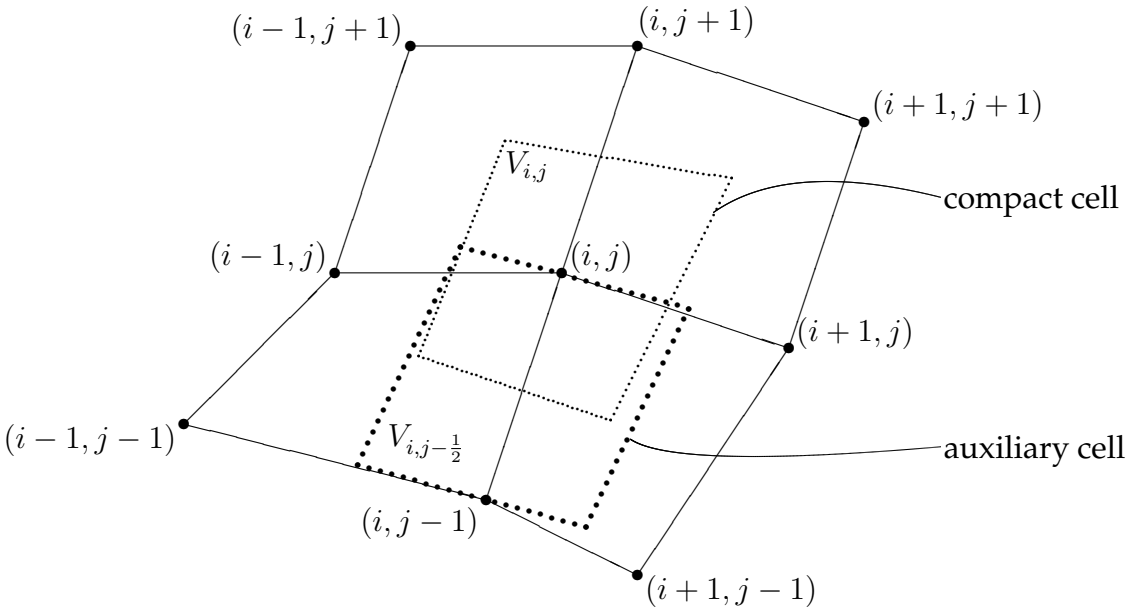

Figure 3.2: Compact and auxiliary cell for central discretization of viscous fluxes and upwind discretization of convective fluxes (2D)

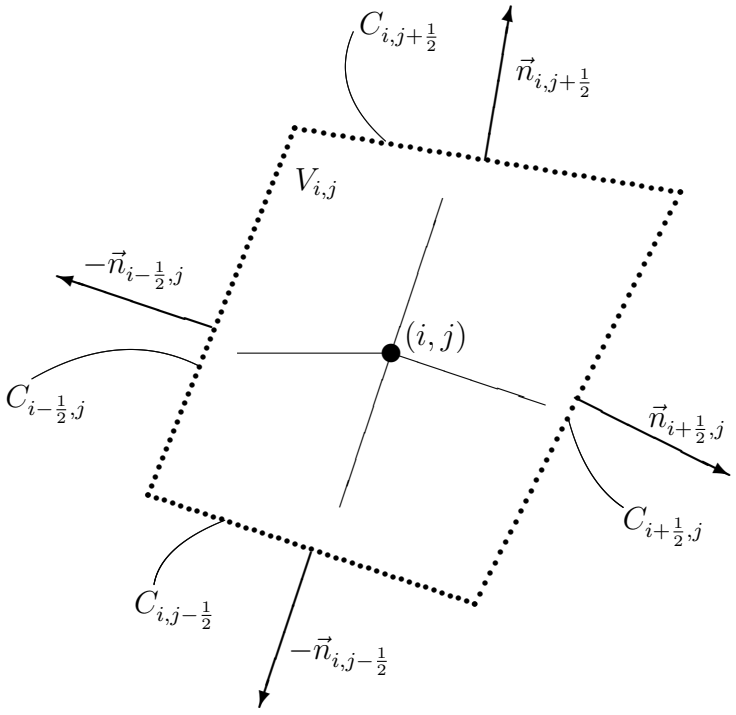

Figure 3.3: Cell faces and outer normal unit vectors of compact cell (2D)

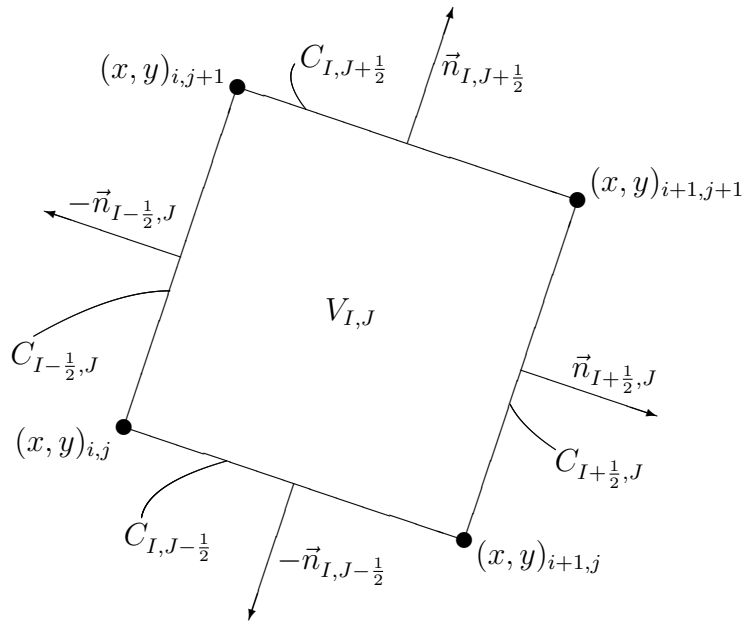

Figure 3.4: Cell faces and outer normal unit vectors of 2D grid cell

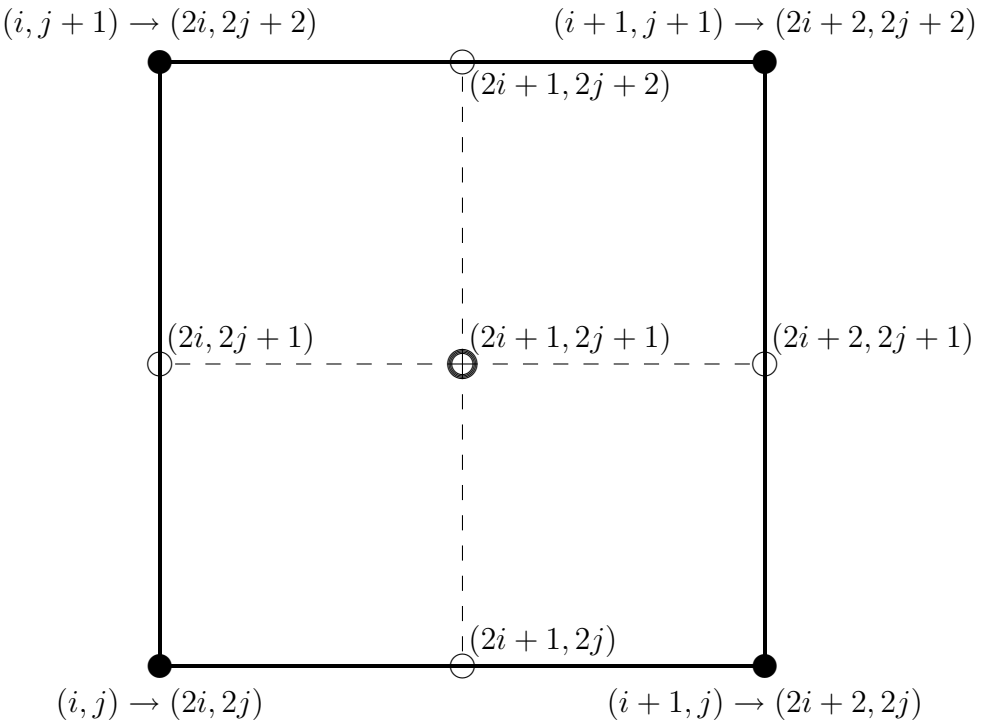

Figure 3.5: Schematic description of coarse and fine grid (2D)

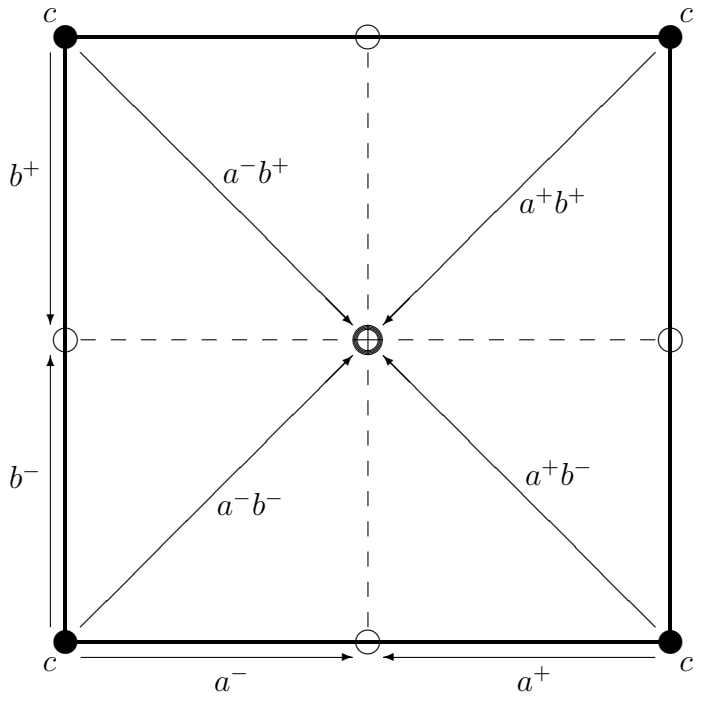

Figure 3.6: Schematic description of generic prolongation (2D)

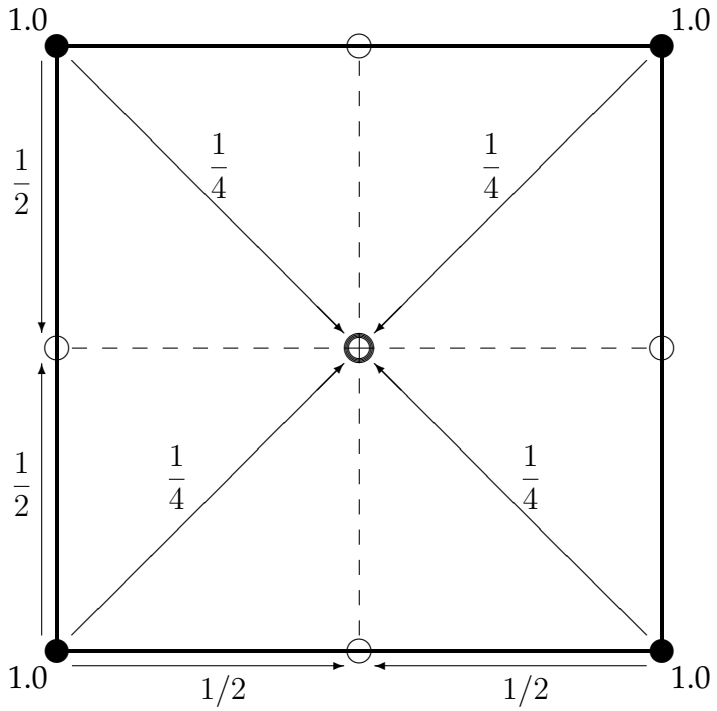

Figure 3.7: Schematic description of bilinear interpolation (2D), i. e. Eqs. (3.111) and (3.112) using $a^- = a^+ = b^- = b^+ = 1/2$ and $c = 1.0$

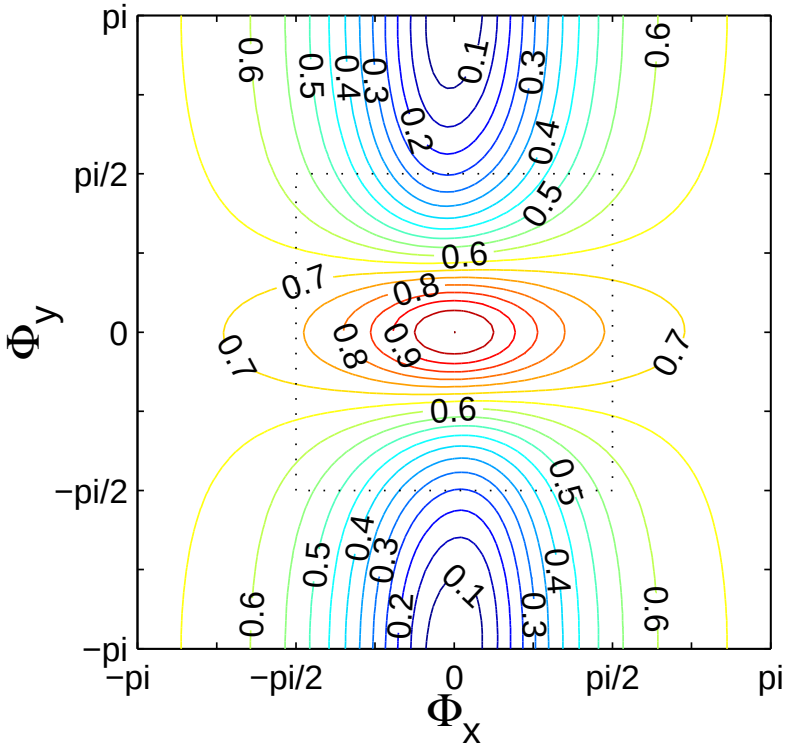

Figure 4.1: Fourier footprint of $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$ with coefficients optimized for central discretization ($\alpha_{\texttt{cent}} = \left[\frac{1}{4}, \frac{1}{6}, \frac{3}{8}, \frac{1}{2}, 1\right] \Rightarrow \mathtt{CFL}_{\mathrm{unsmoothed}} = 3.75, \mathtt{CFL}_{\mathrm{smoothed}} = 7.5$) applied to Eq. (4.1) [$u = v = 1$; $\Delta x = \Delta y = 1$; $\mu = 1$; $S = 0$]

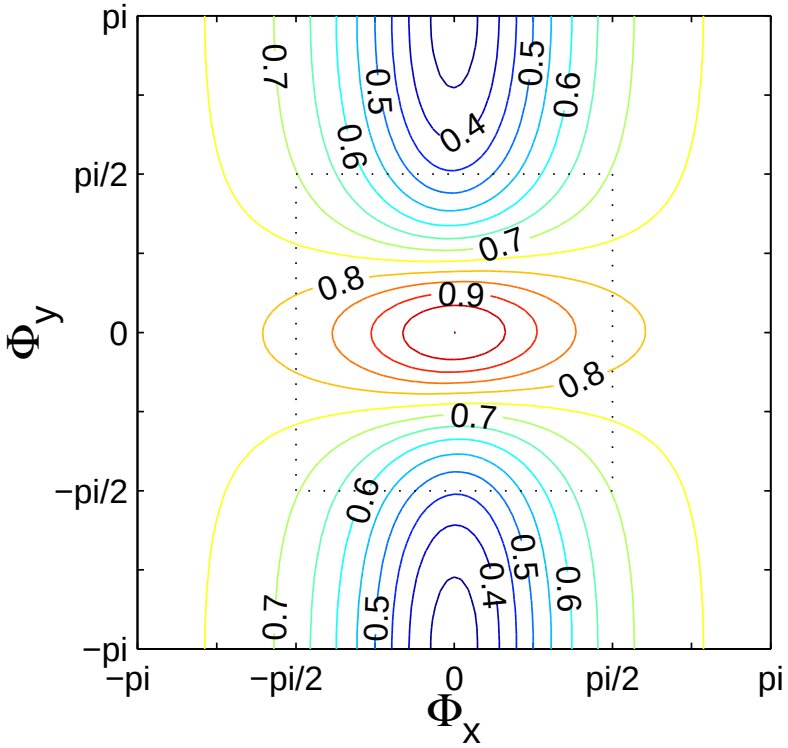

Figure 4.2: Fourier footprint of $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$ with coefficients optimized for upwind discretization ($\alpha_{\texttt{upw}} = [0.0533, 0.1263, 0.2375, 0.4414, 1] \Rightarrow \mathtt{CFL}_{\mathrm{unsmoothed}} = 2.5, \mathtt{CFL}_{\mathrm{smoothed}} = 5.0$) applied to Eq. (4.1) [$u = v = 1$; $\Delta x = \Delta y = 1$; $\mu = 1$; $S = 0$]

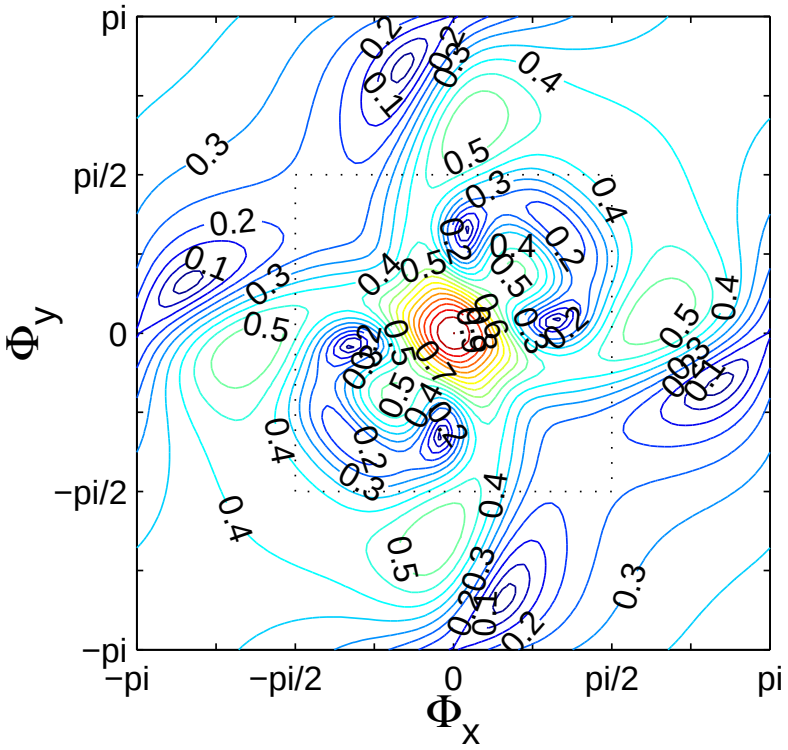

Figure 4.3: Fourier footprint of $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = v = 1$; $\Delta x = \Delta y = 1$; $\mu = 0$; $S = 0$]: convection equation with $\Delta x/\Delta y = 1$ and $(v\Delta x)/(u\Delta y) = 1$

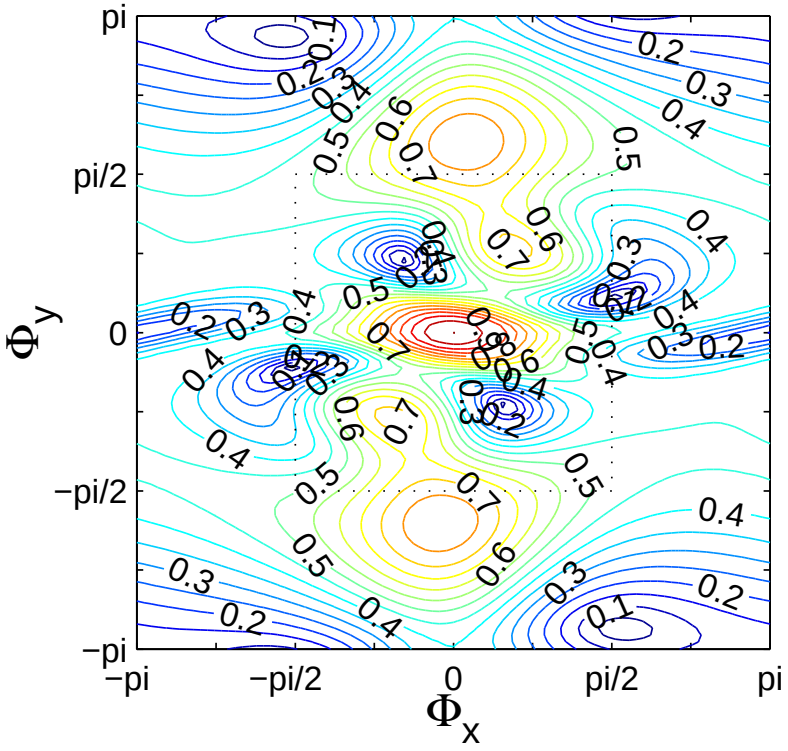

Figure 4.4: Fourier footprint of $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = v = 1$; $\Delta x = 1$; $\Delta y = 0.2$; $\mu = 0$; $S = 0$]: convection equation with $\Delta x/\Delta y = 5$ and $(v\Delta x)/(u\Delta y) = 5$

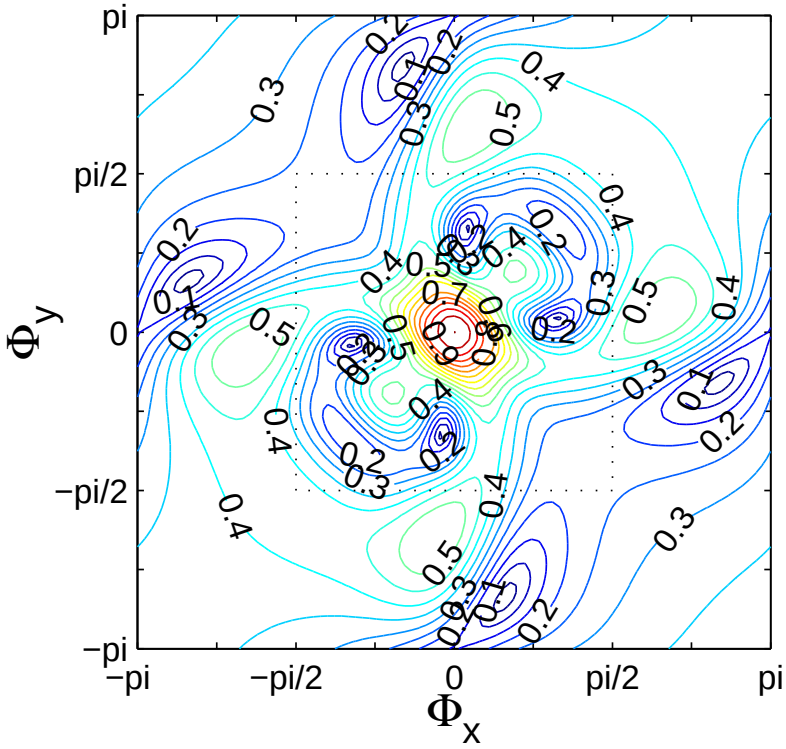

Figure 4.5: Fourier footprint of $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 1$; $v = 0.2$; $\Delta x = 1$; $\Delta y = 0.2$; $\mu = 0$; $S = 0$]: convection equation with $\Delta x/\Delta y = 5$ and $(v\Delta x)/(u\Delta y) = 1$

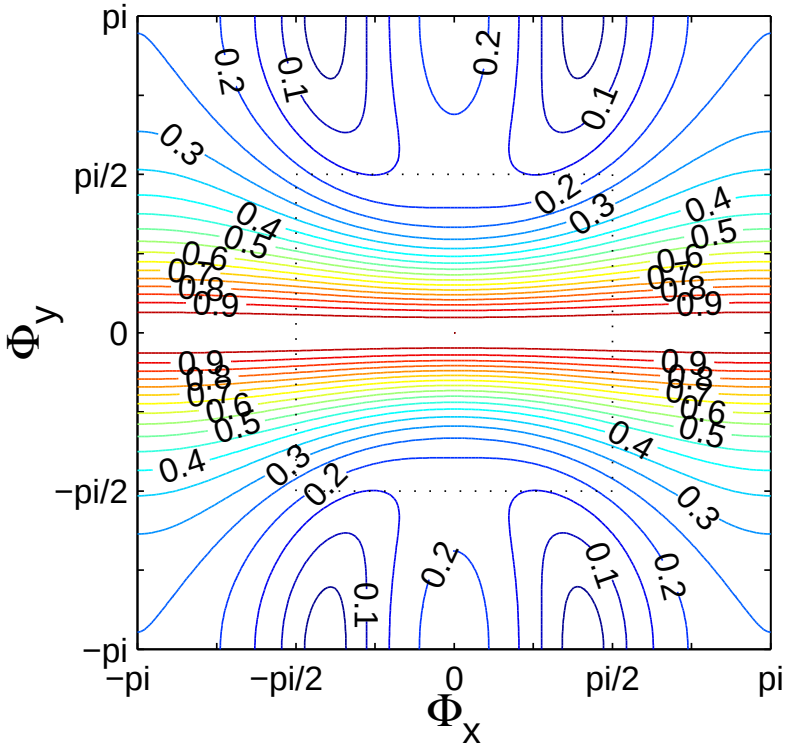

Figure 4.6: Fourier footprint of $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 0$]: convection diffusion equation with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$

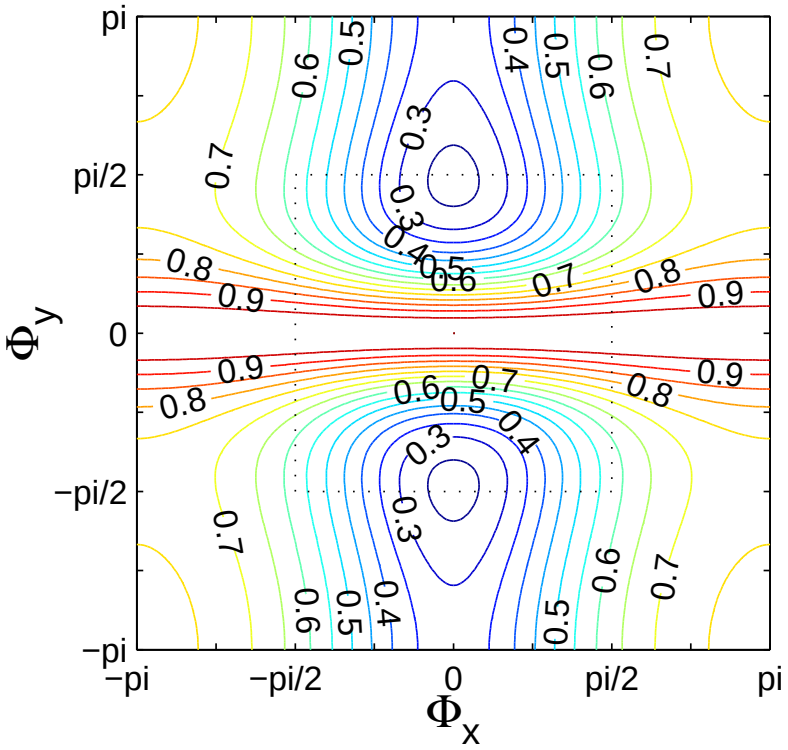

Figure 4.7: Fourier footprint of $g^{(q=5),\,\mathrm{FG}}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 0$]: convection diffusion equation with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; $\varepsilon_{\mathrm{MG}} = 0.2$

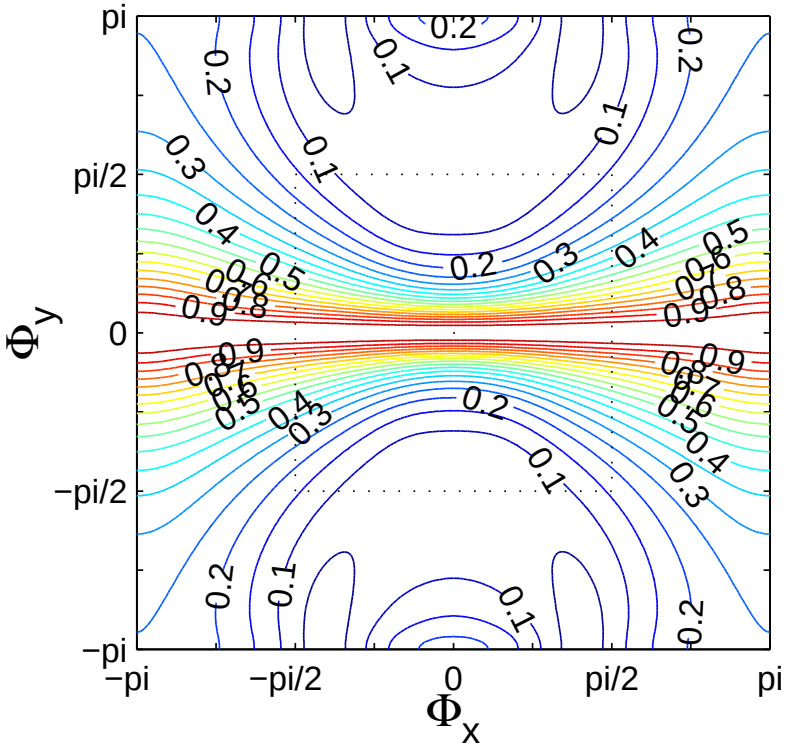

Figure 4.8: Fourier footprint of $g^{(q=5),\,\mathrm{2G}}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 0$]: convection diffusion equation with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 2 multigrid level ($\mathcal{P}^H_{2H} = 1\ \forall H$)

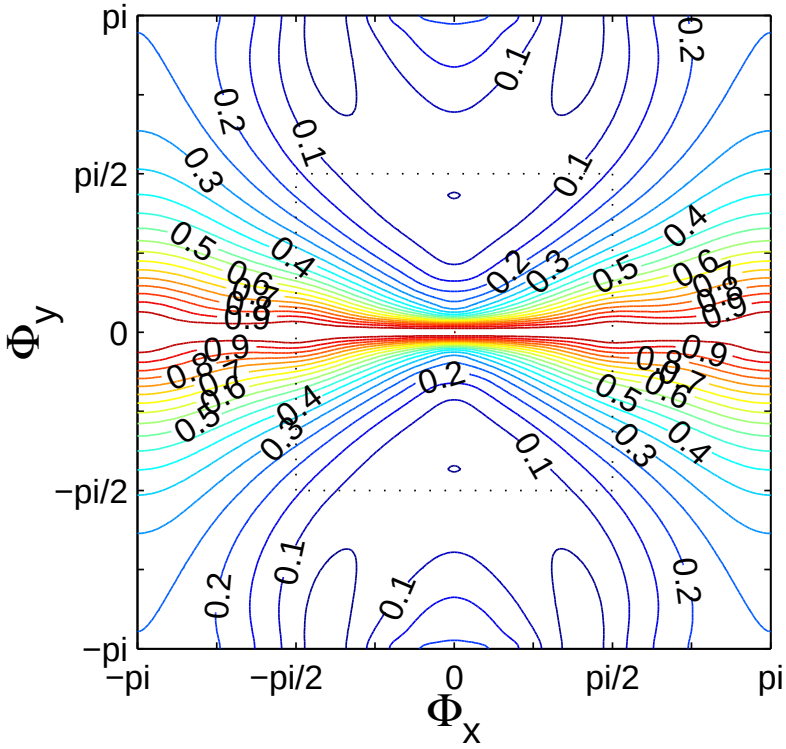

Figure 4.9: Fourier footprint of $g^{(q=5),\,\mathrm{MG}}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\texttt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 0$]: convection diffusion equation with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 3 multigrid level ($\mathcal{P}^H_{2H} = 1\ \forall H$)

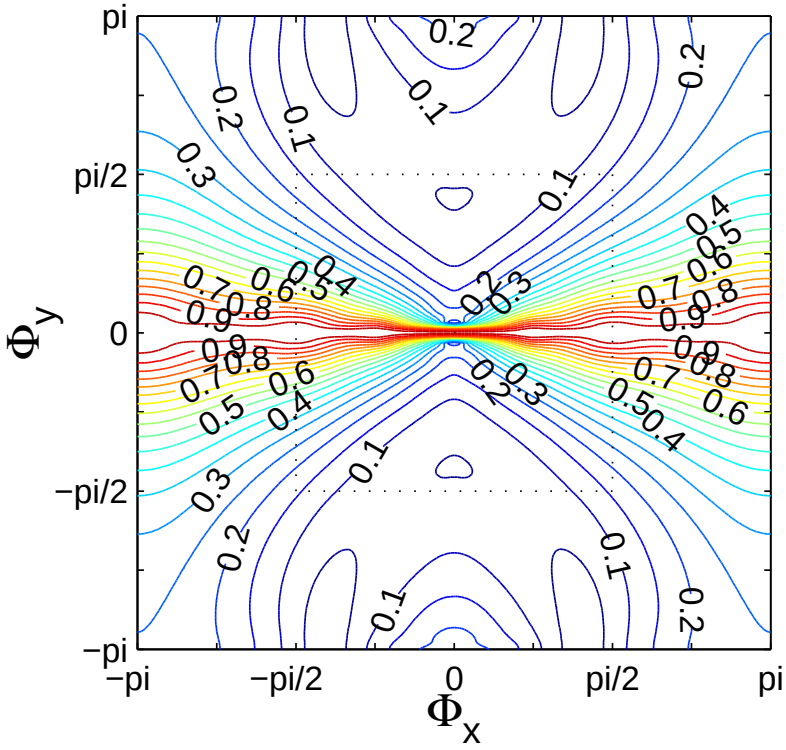

Figure 4.10: Fourier footprint of $g^{(q=5),\,\mathrm{MG}}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\texttt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 0$]: convection diffusion equation with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 4 multigrid level ($\mathcal{P}^H_{2H} = 1\ \forall H$)

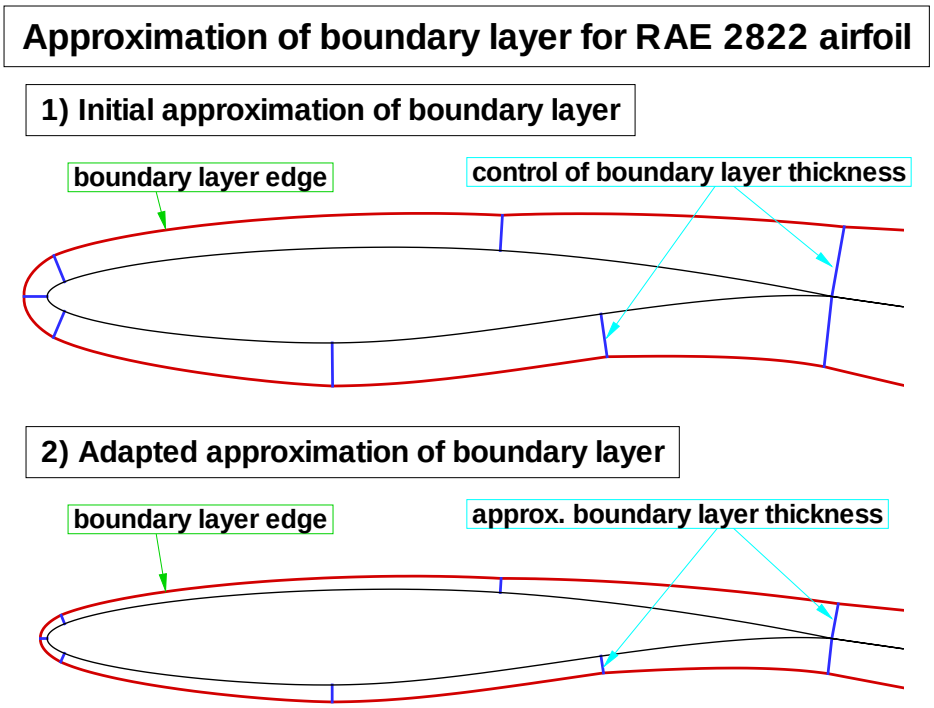

Figure 5.1: Basic grid construction for approximation of boundary layer at RAE 2822 airfoil

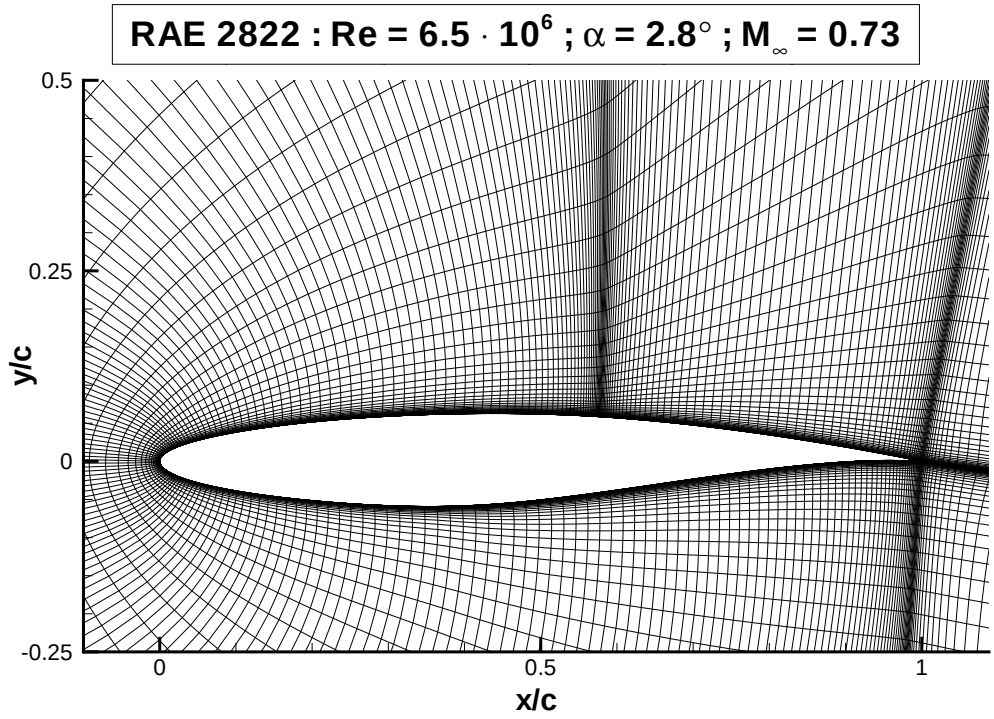

Figure 5.2: Computational grid for RAE 2822 airfoil at $\mathrm{Re} = 6.5 \cdot 10^6$

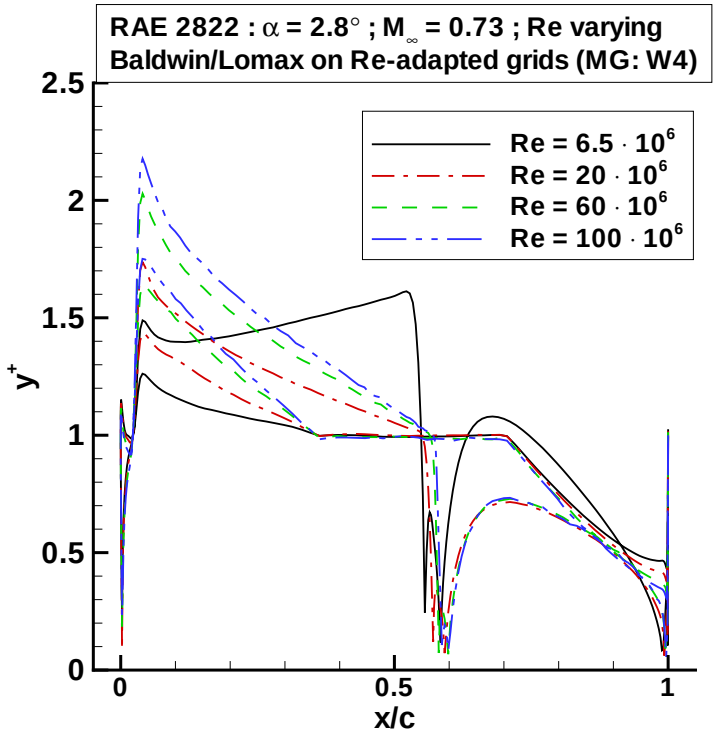

Figure 5.3: y^+_{wall} distribution; RAE 2822 airfoil, RANS calculation using Baldwin/Lomax turbulence model, varying Reynolds number; 4 multigrid level, W-cycle

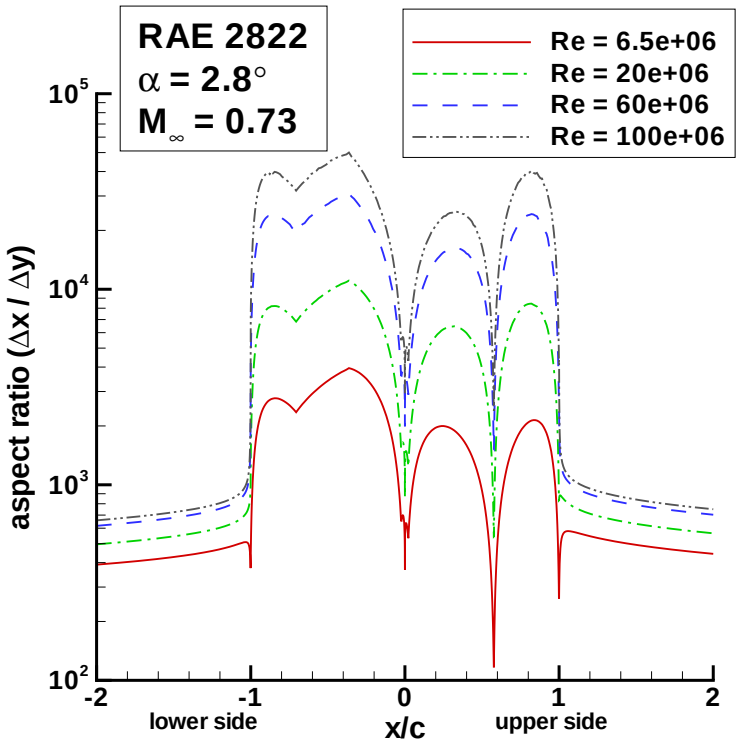

Figure 5.4: Aspect ratio of first grid cells on the wall of RAE 2822 airfoil in Re–adapted grids, varying Reynolds number

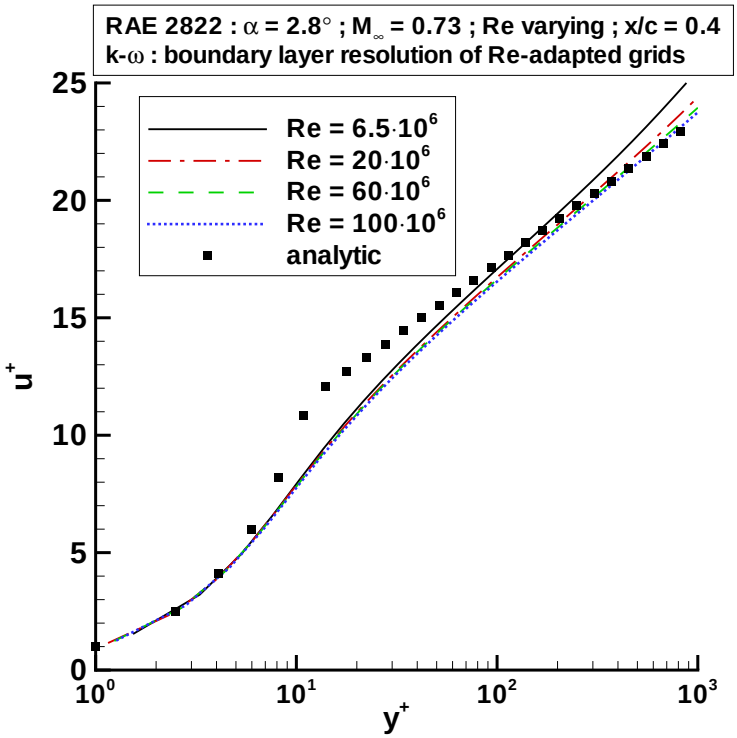

Figure 5.5: Dimensionless velocity profile at 40 % chord length (upper side) within boundary layer for Re–adapted grids around RAE 2822 airfoil, varying Reynolds number

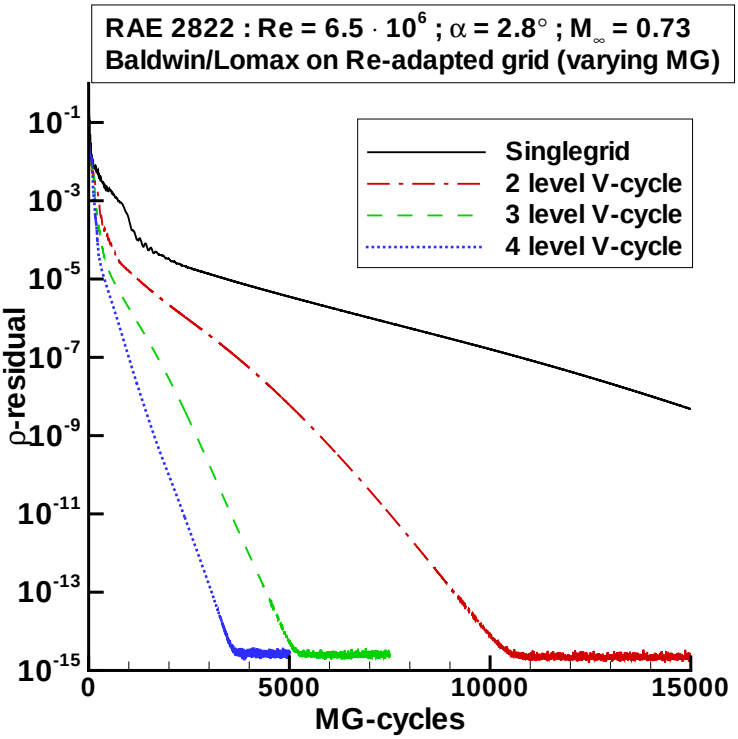

Figure 5.6: Convergence of density residual; RAE 2822 airfoil, RANS calculation using Baldwin/Lomax turbulence model, Re $= 6.5 \cdot 10^6$, varying number of multigrid levels in use, V–cycle

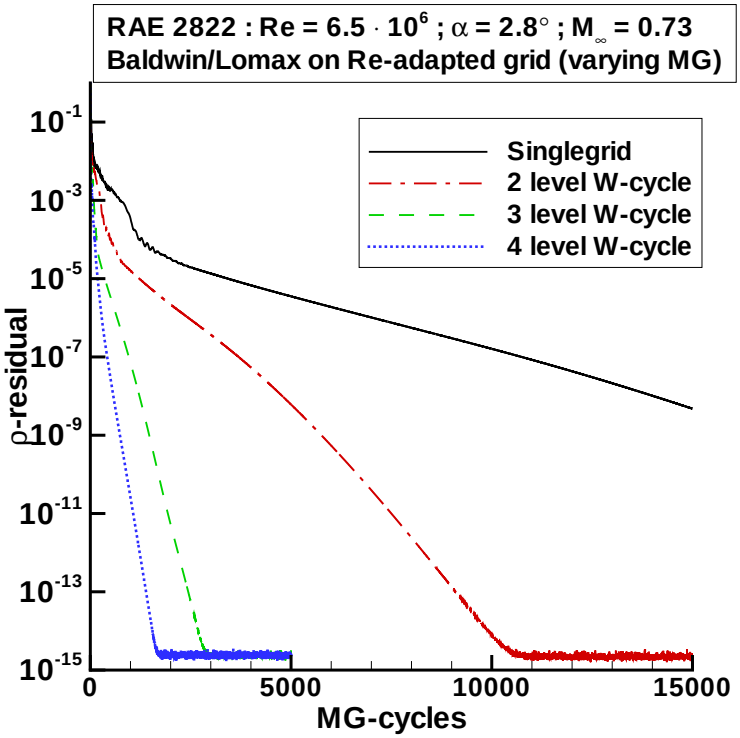

Figure 5.7: Convergence of density residual; RAE 2822 airfoil, RANS calculation using Baldwin/Lomax turbulence model, Re $= 6.5 \cdot 10^6$, varying number of multigrid levels in use, W–cycle

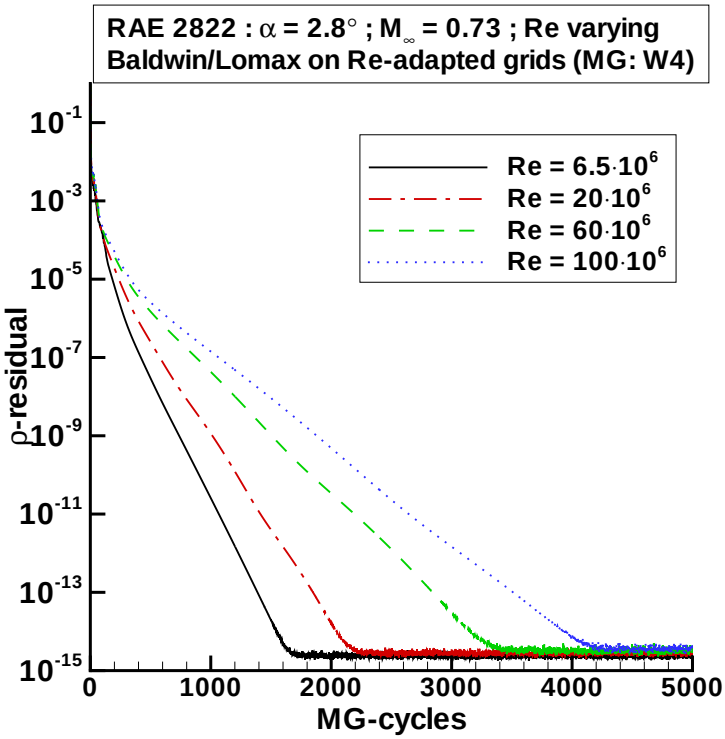

Figure 5.8: Convergence of density residual; RAE 2822 airfoil, RANS calculation using Baldwin/Lomax turbulence model, varying Reynolds number; 4 multigrid level, W–cycle

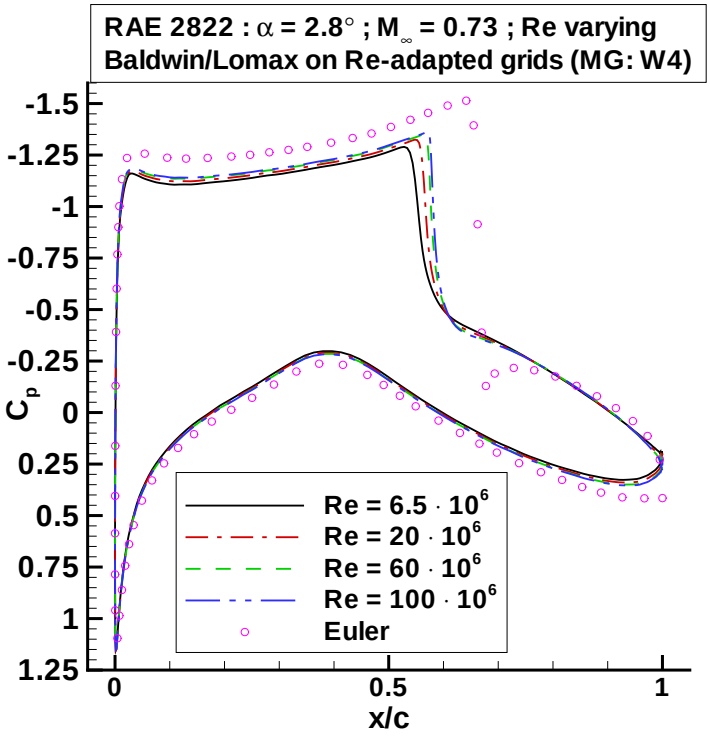

Figure 5.9: C_p distribution; RAE 2822 airfoil, RANS calculation using Baldwin/Lomax turbulence model, varying Reynolds number; 4 multigrid level, W–cycle

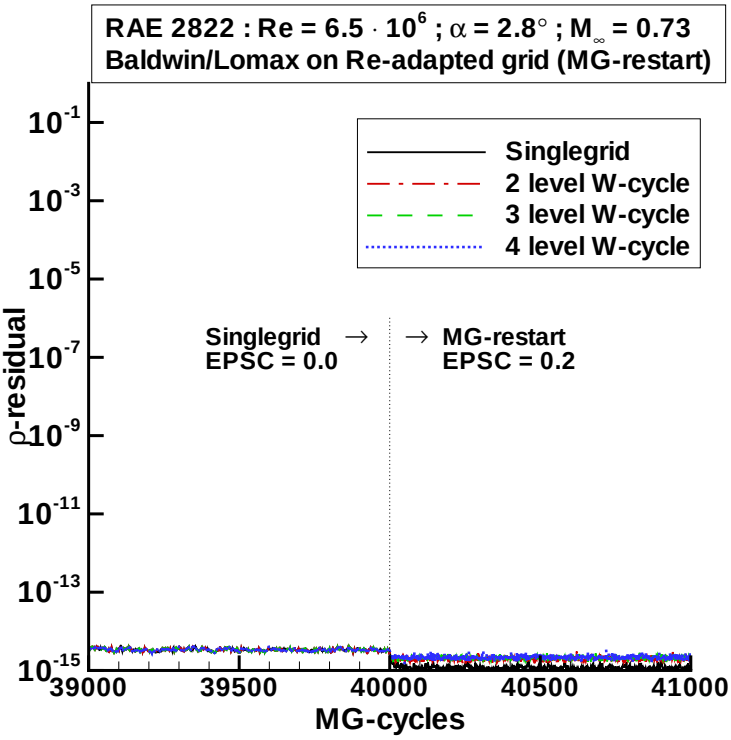

Figure 5.10: Convergence of density residual; RAE 2822 airfoil, RANS calculation using Baldwin/Lomax turbulence model, $\mathrm{Re} = 6.5 \cdot 10^6$, restart from singlegrid solution (obtained using $\mathtt{EPSC} = 0.0$) with varying number of multigrid levels in use

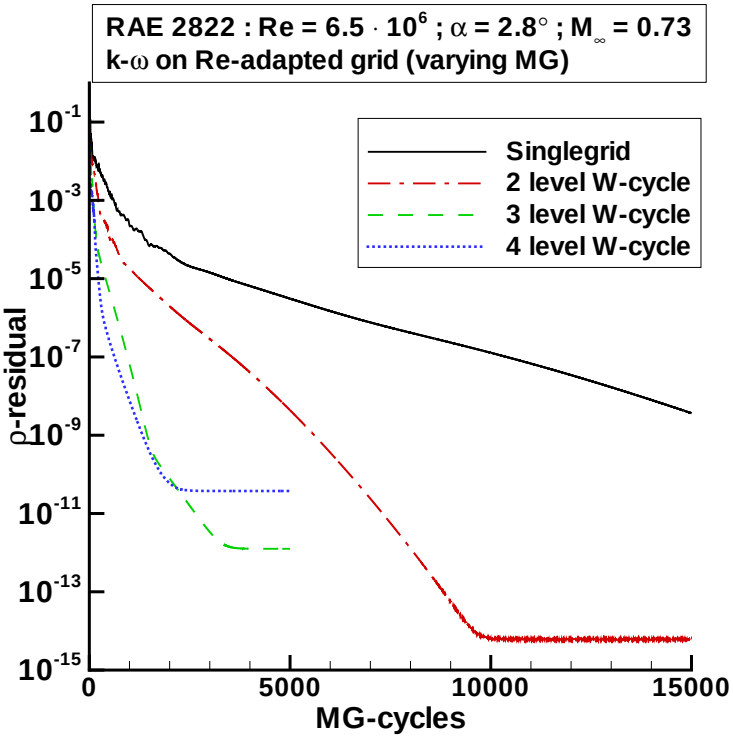

Figure 5.11: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 6.5 \cdot 10^6$, varying number of multigrid levels in use

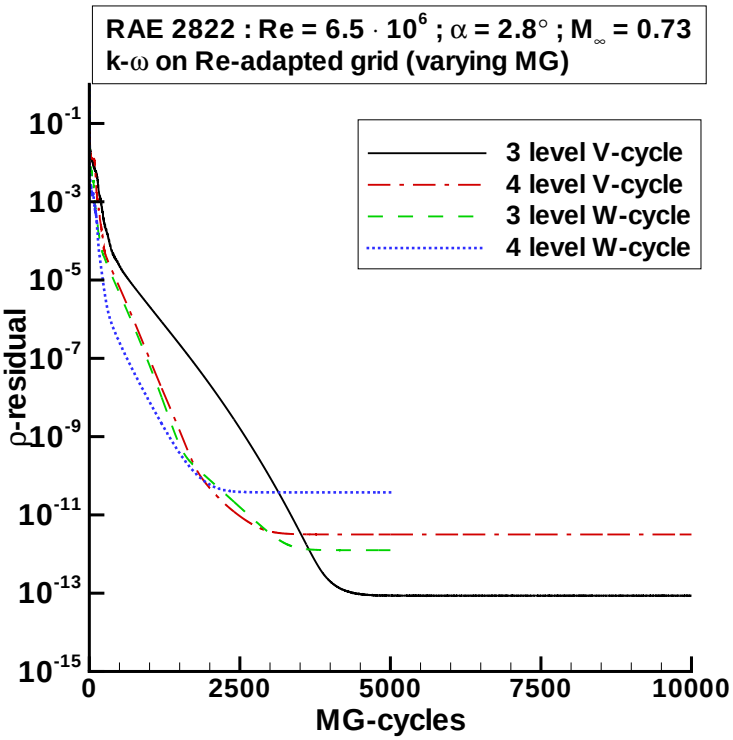

Figure 5.12: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 6.5 \cdot 10^6$, varying number of multigrid levels and type of multigrid cycle in use

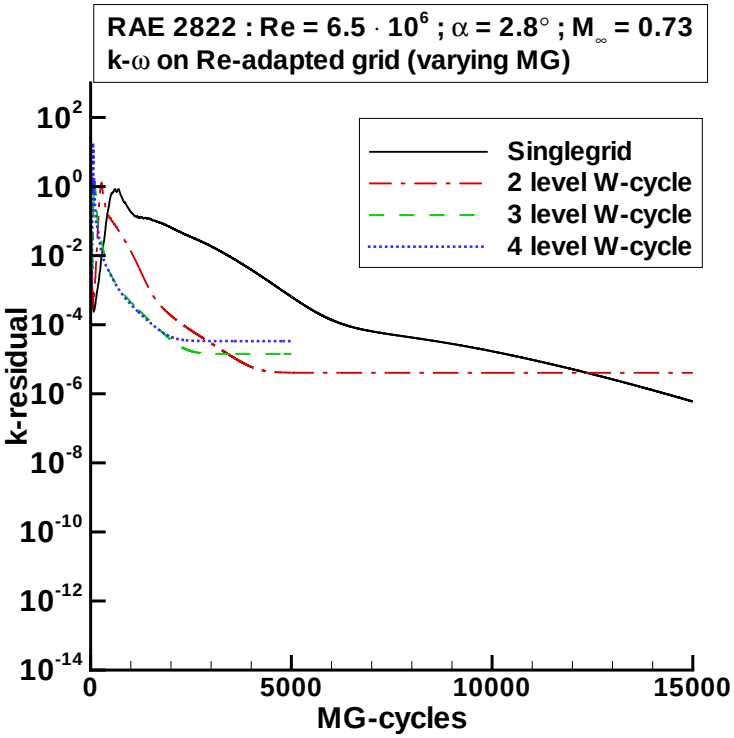

Figure 5.13: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 6.5 \cdot 10^6$, varying number of multigrid levels in use

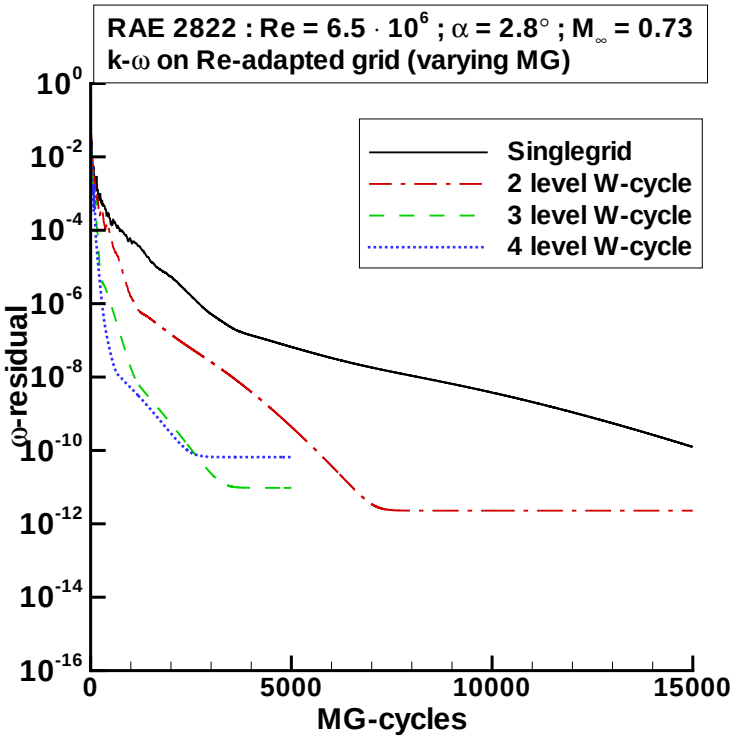

Figure 5.14: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 6.5 \cdot 10^6$, varying number of multigrid levels in use

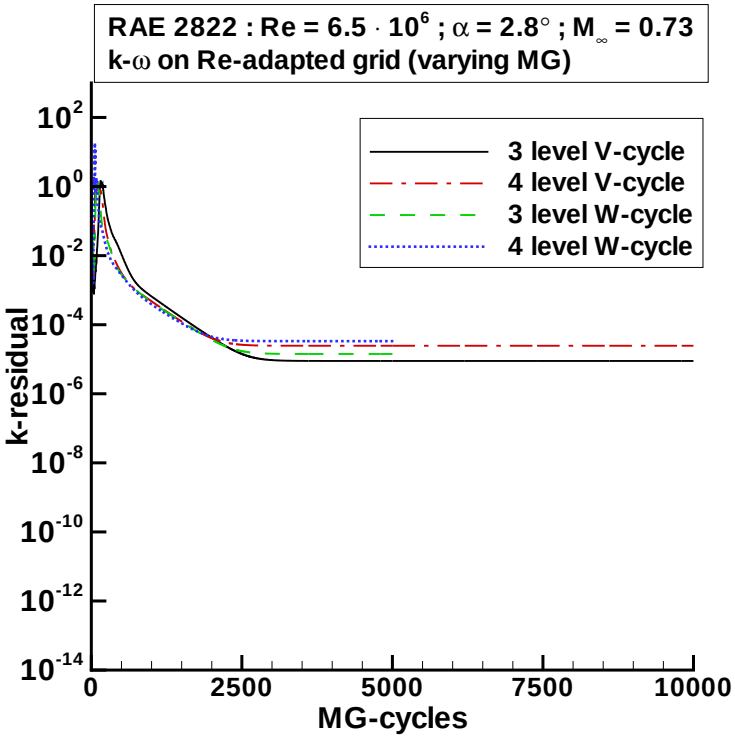

Figure 5.15: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 6.5 \cdot 10^6$, varying number of multigrid levels and type of multigrid cycle in use

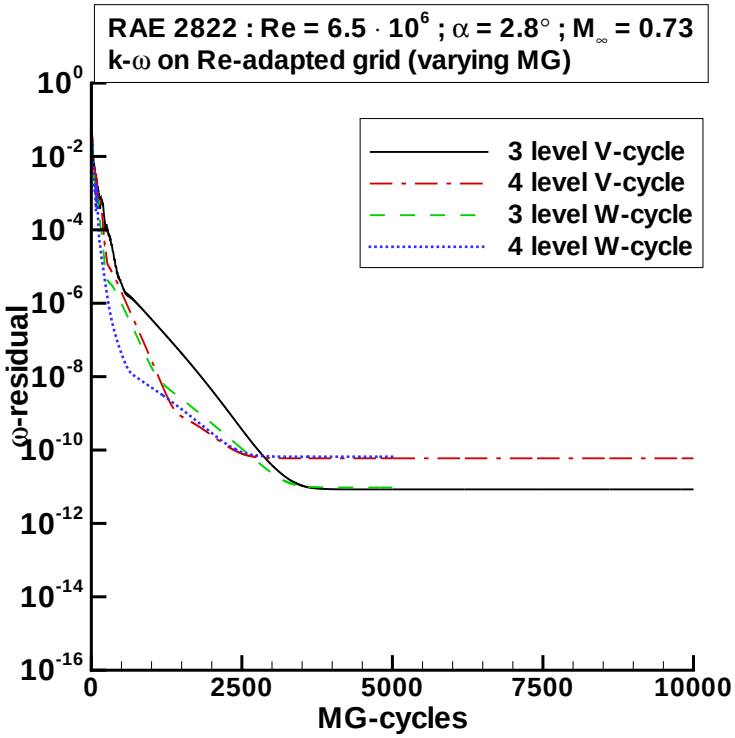

Figure 5.16: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 6.5 \cdot 10^6$, varying number of multigrid levels and type of multigrid cycle in use

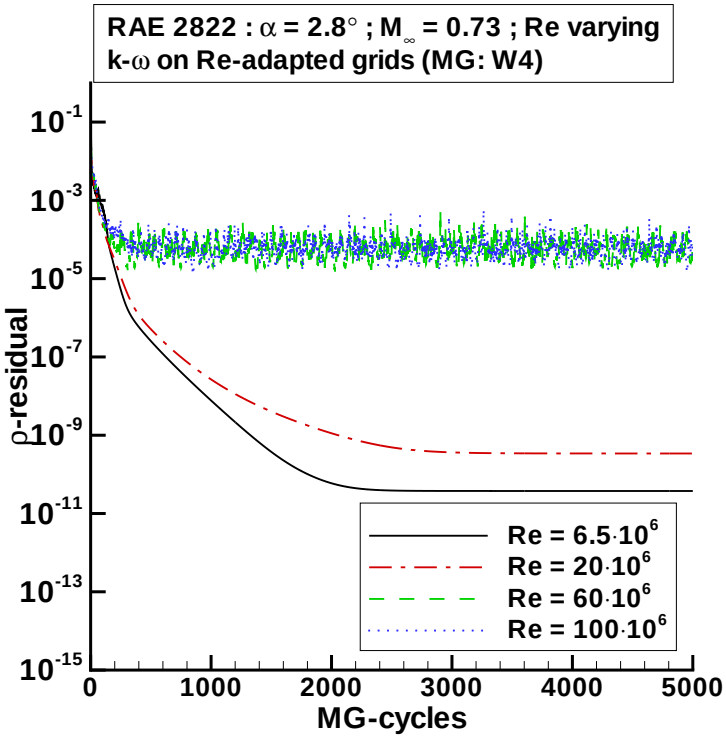

Figure 5.17: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, varying Reynolds number; 4 multigrid level, W-cycle

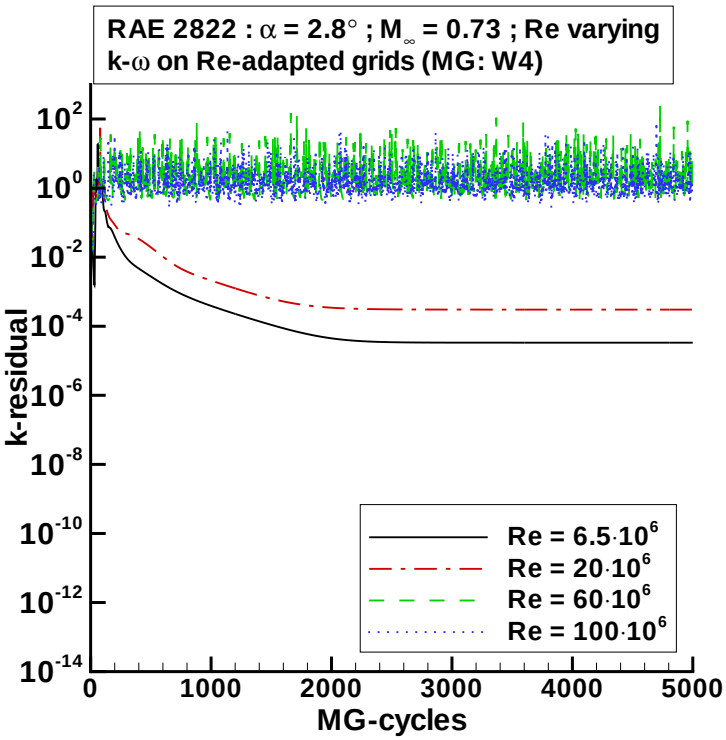

Figure 5.18: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, varying Reynolds number; 4 multigrid level, W-cycle

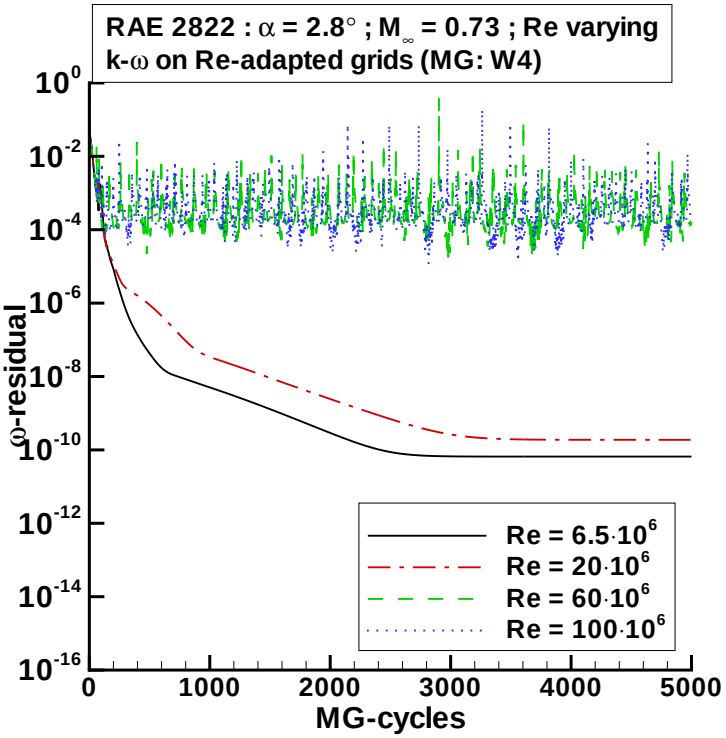

Figure 5.19: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, varying Reynolds number; 4 multigrid level, W-cycle

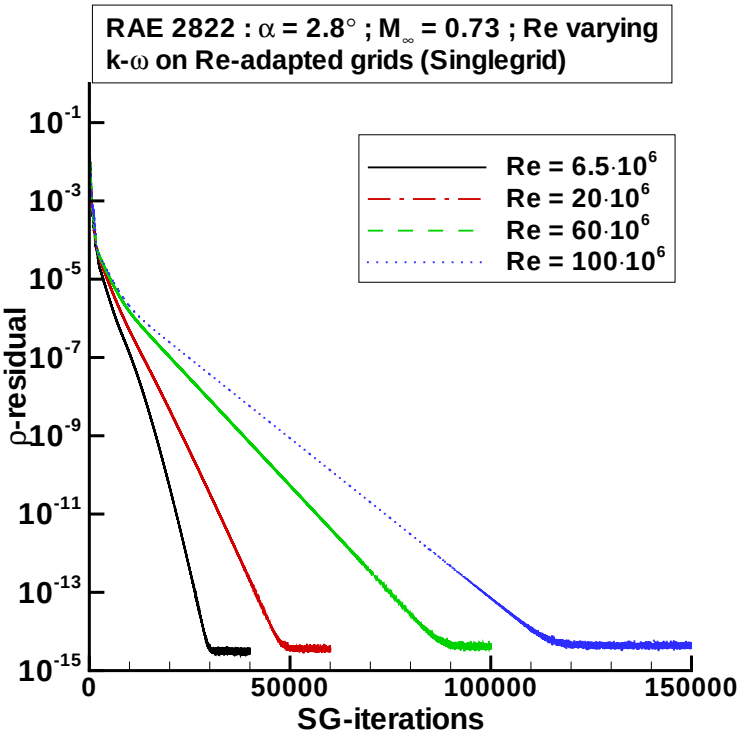

Figure 5.20: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, varying Reynolds number; single grid

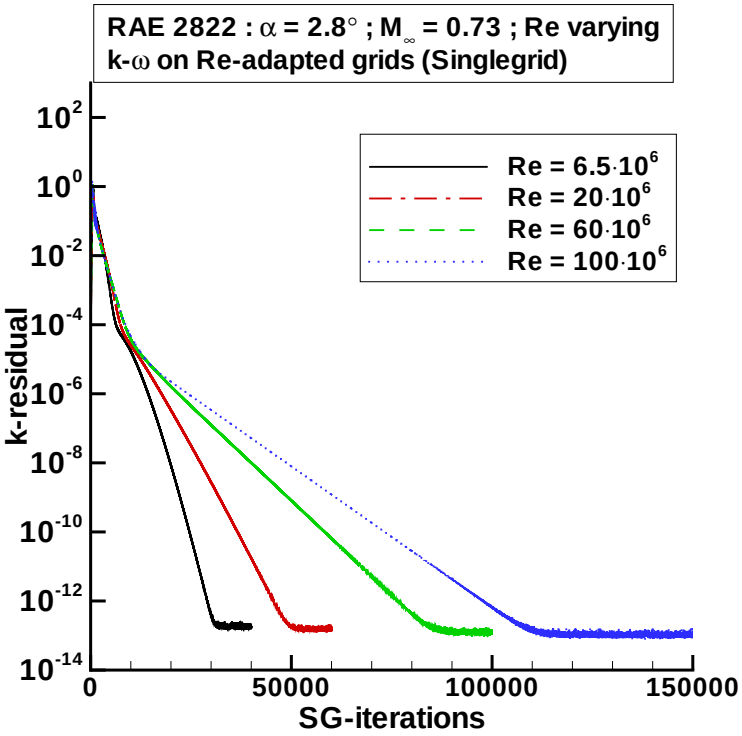

Figure 5.21: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, varying Reynolds number; single grid

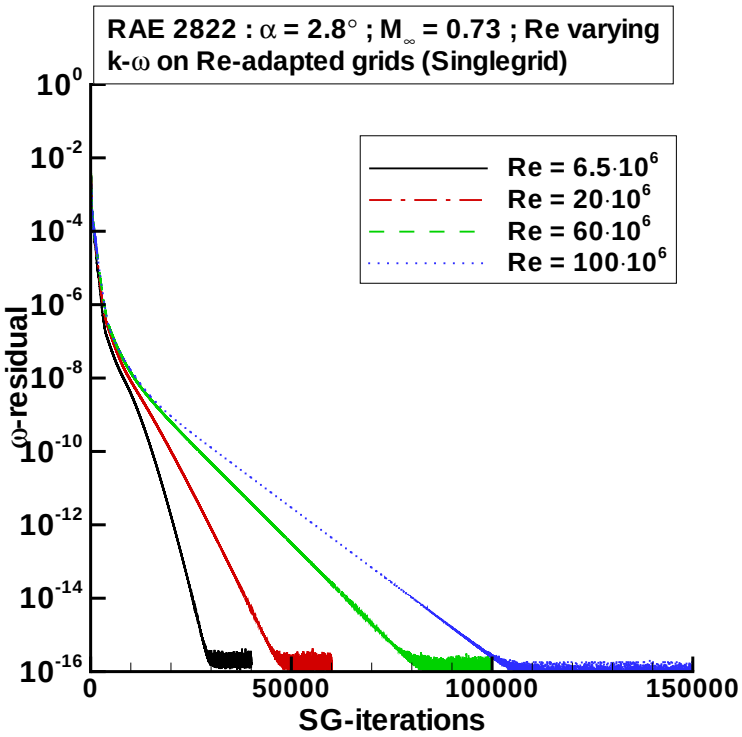

Figure 5.22: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, varying Reynolds number; single grid

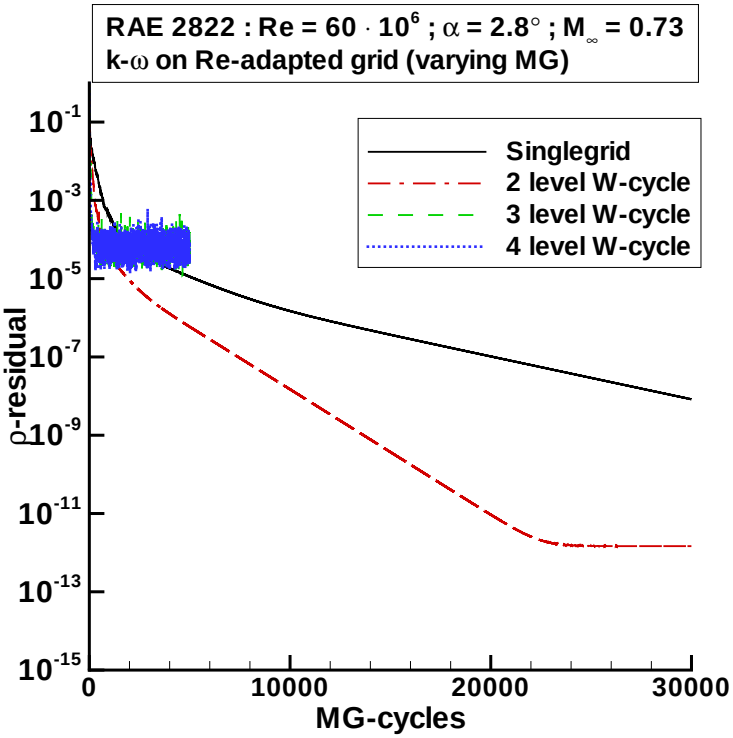

Figure 5.23: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 60 \cdot 10^6$, varying number of multigrid levels in use

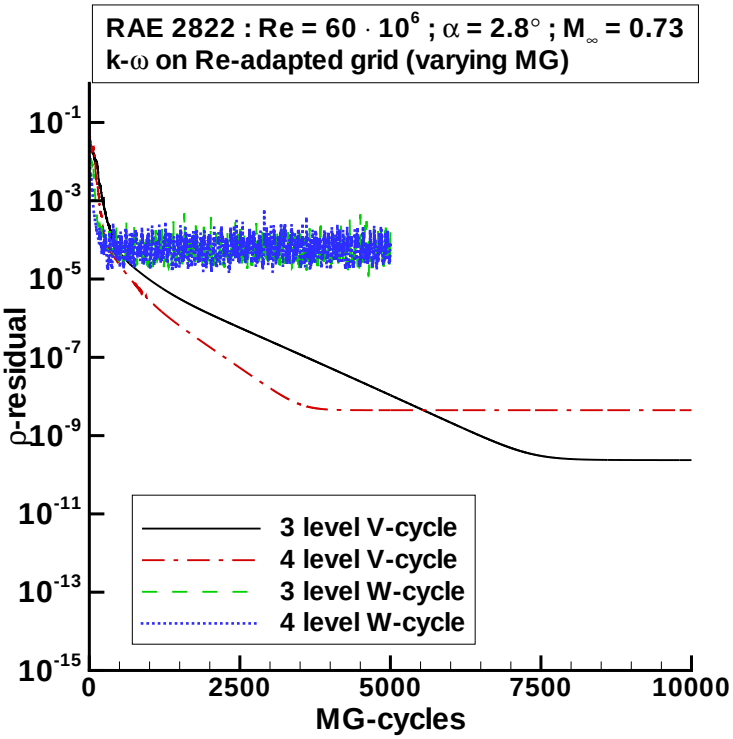

Figure 5.24: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 60 \cdot 10^6$, varying number of multigrid levels and type of multigrid cycle in use

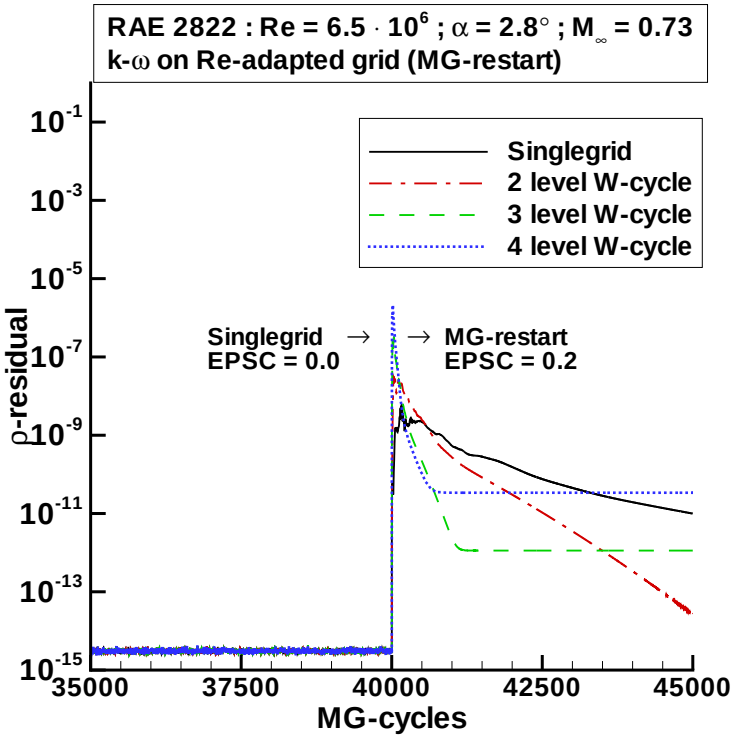

Figure 5.25: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 6.5 \cdot 10^6$, restart from singlegrid solution (obtained using EPSC $= 0.0$) with varying number of multigrid levels in use

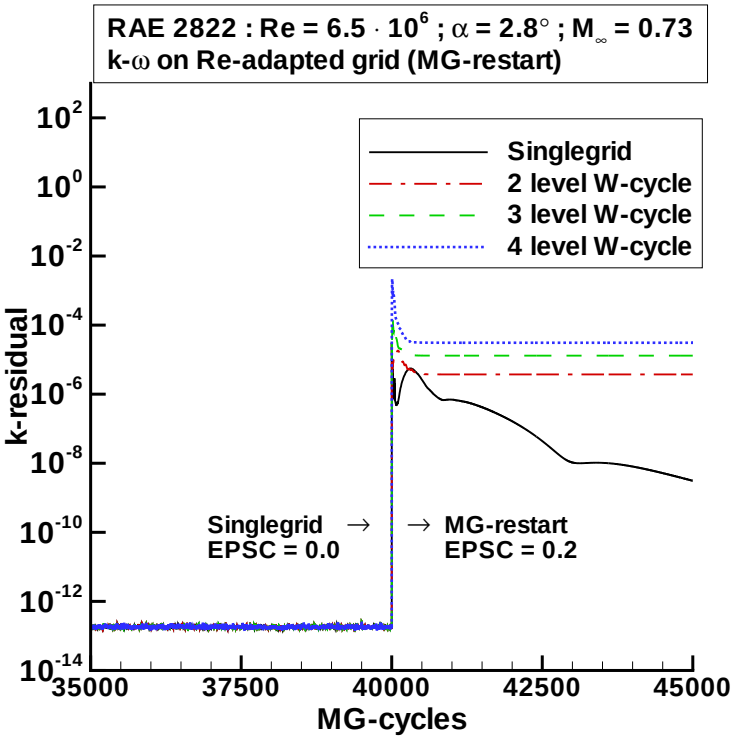

Figure 5.26: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 6.5 \cdot 10^6$, restart from singlegrid solution (obtained using EPSC $= 0.0$) with varying number of multigrid levels in use

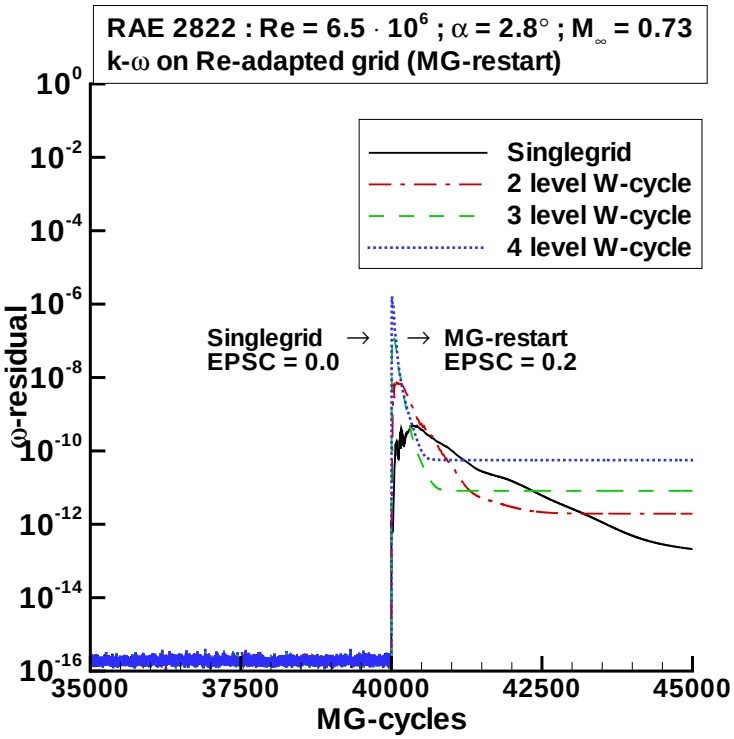

Figure 5.27: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, $\mathrm{Re} = 6.5 \cdot 10^6$, restart from singlegrid solution (obtained using $\texttt{EPSC} = 0.0$) with varying number of multigrid levels in use

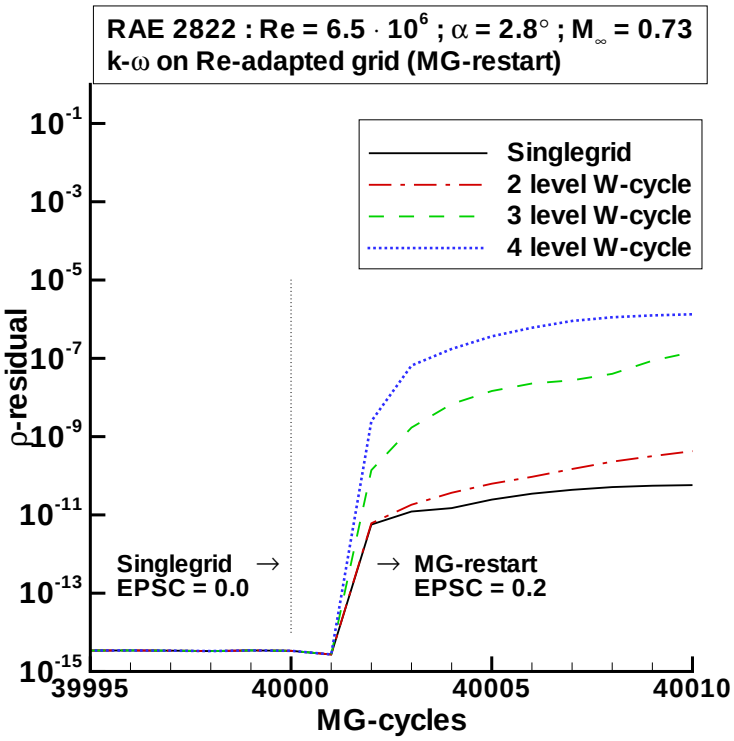

Figure 5.28: Close–up of convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, $\mathrm{Re} = 6.5 \cdot 10^6$, restart from singlegrid solution (obtained using $\texttt{EPSC} = 0.0$) with varying number of multigrid levels in use

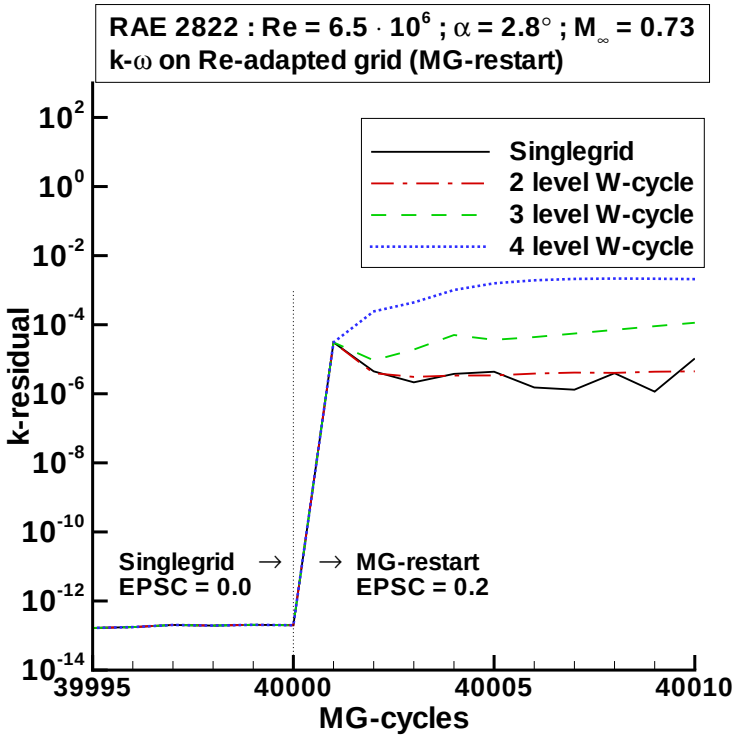

Figure 5.29: Close–up of convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 6.5 \cdot 10^6$, restart from singlegrid solution (obtained using EPSC $=$ 0.0) with varying number of multigrid levels in use

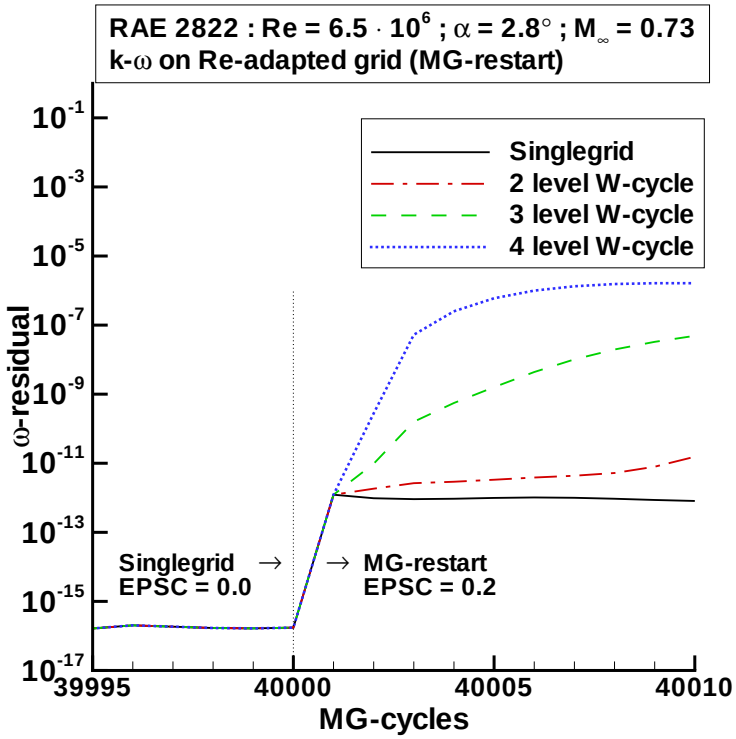

Figure 5.30: Close–up of convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 6.5 \cdot 10^6$, restart from singlegrid solution (obtained using EPSC $=$ 0.0) with varying number of multigrid levels in use

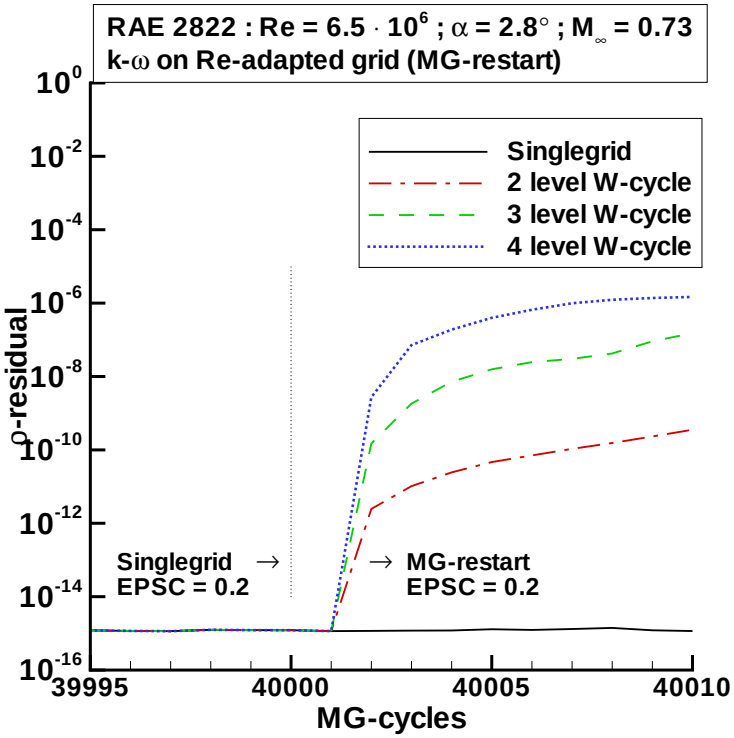

Figure 5.31: Close–up of convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, $\mathrm{Re} = 6.5 \cdot 10^6$, restart from singlegrid solution (obtained using $\texttt{EPSC} = 0.2$) with varying number of multigrid levels in use

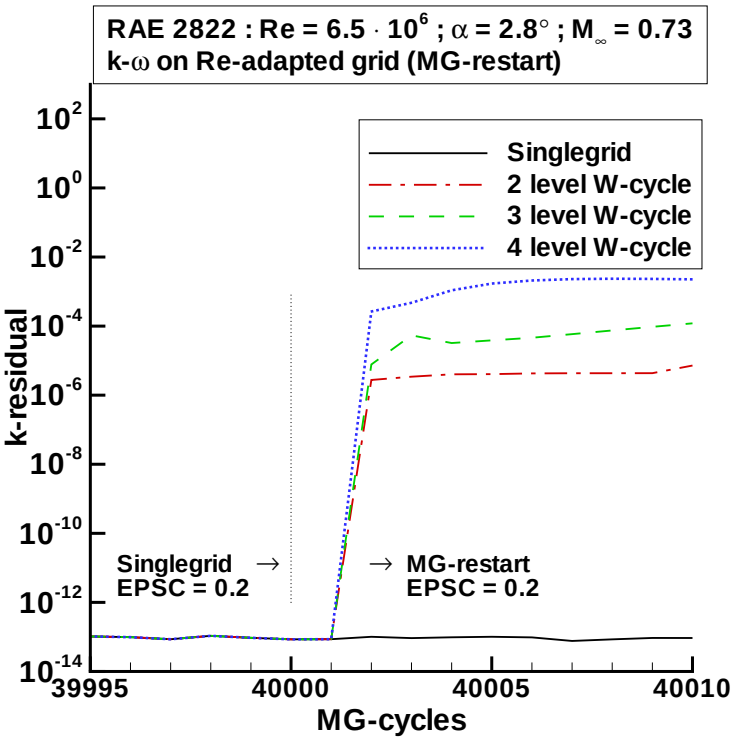

Figure 5.32: Close–up of convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, $\mathrm{Re} = 6.5 \cdot 10^6$, restart from singlegrid solution (obtained using $\texttt{EPSC} = 0.2$) with varying number of multigrid levels in use

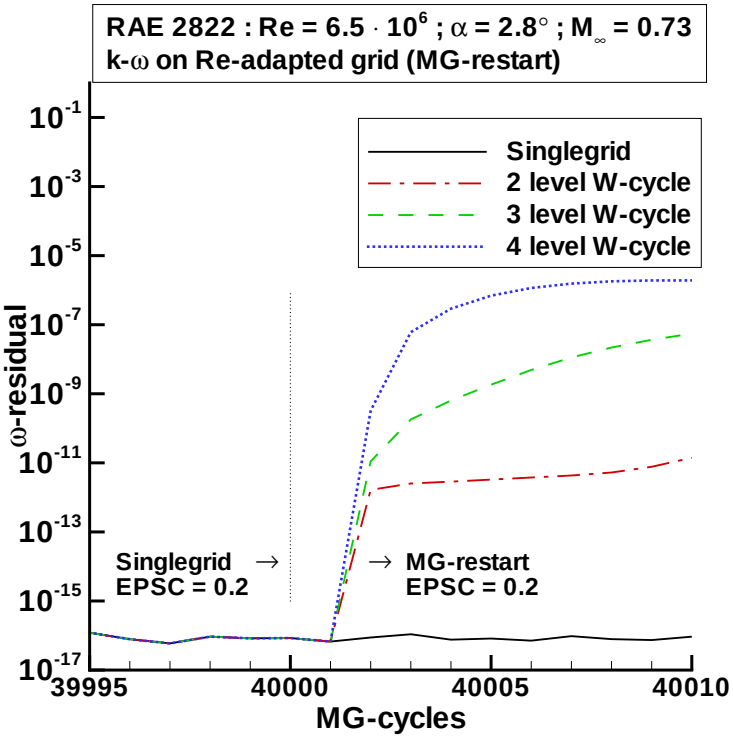

Figure 5.33: Close–up of convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 6.5 \cdot 10^6$, restart from singlegrid solution (obtained using EPSC $= 0.2$) with varying number of multigrid levels in use

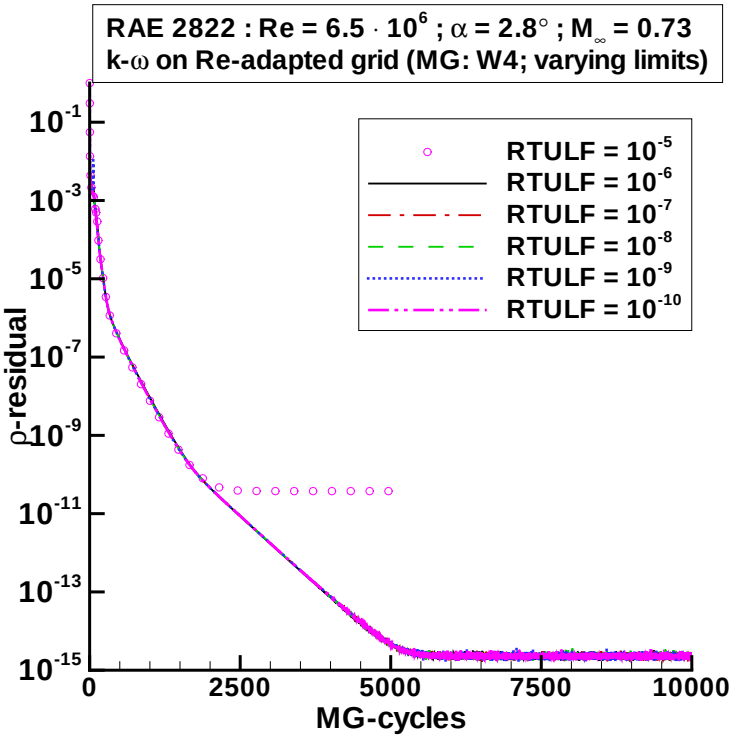

Figure 5.34: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 6.5 \cdot 10^6$; 4 multigrid level, W–cycle, variation of limiting factor of turbulence variables

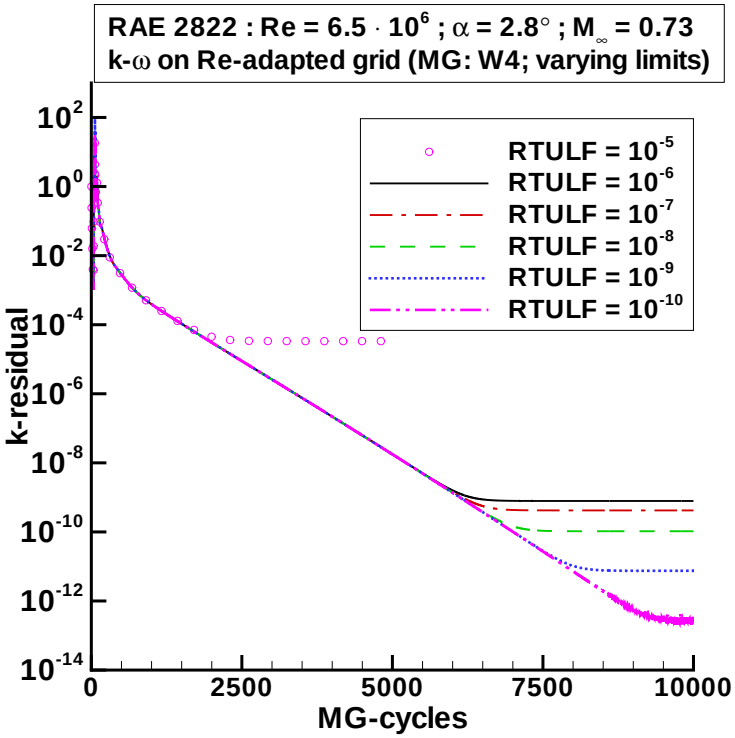

Figure 5.35: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 6.5 \cdot 10^6$; 4 multigrid level, W–cycle, variation of limiting factor of turbulence variables

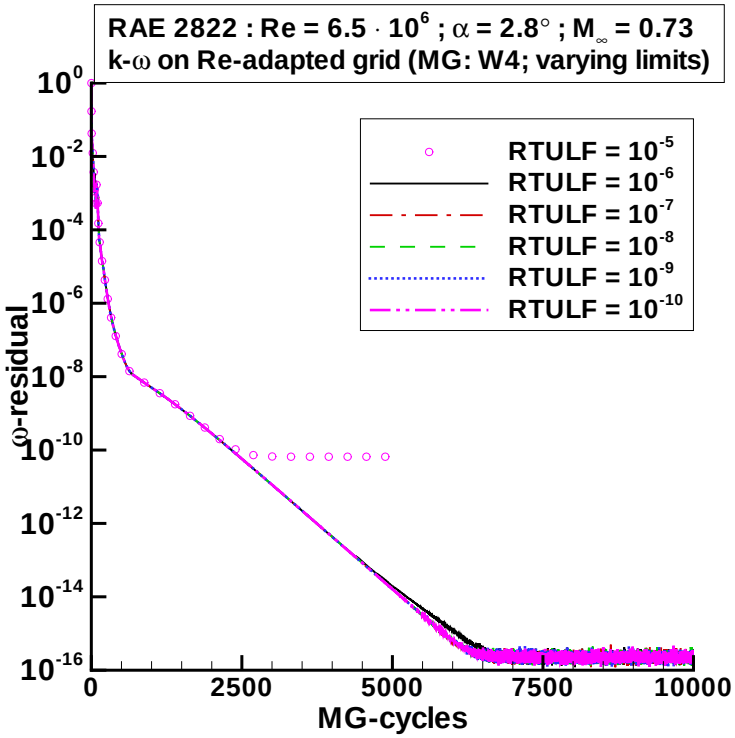

Figure 5.36: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 6.5 \cdot 10^6$; 4 multigrid level, W–cycle, variation of limiting factor of turbulence variables

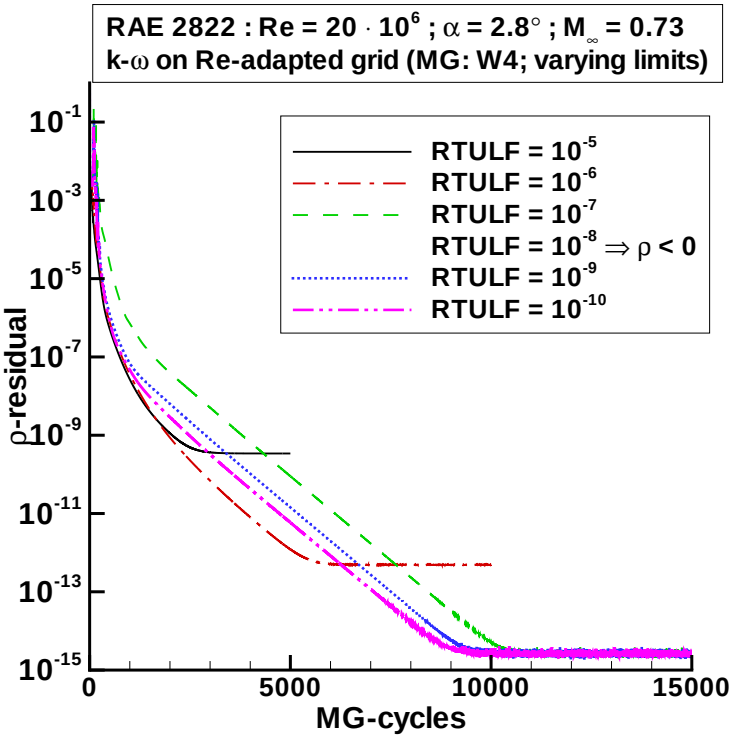

Figure 5.37: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 20 \cdot 10^6$; 4 multigrid level, W–cycle, variation of limiting factor of turbulence variables

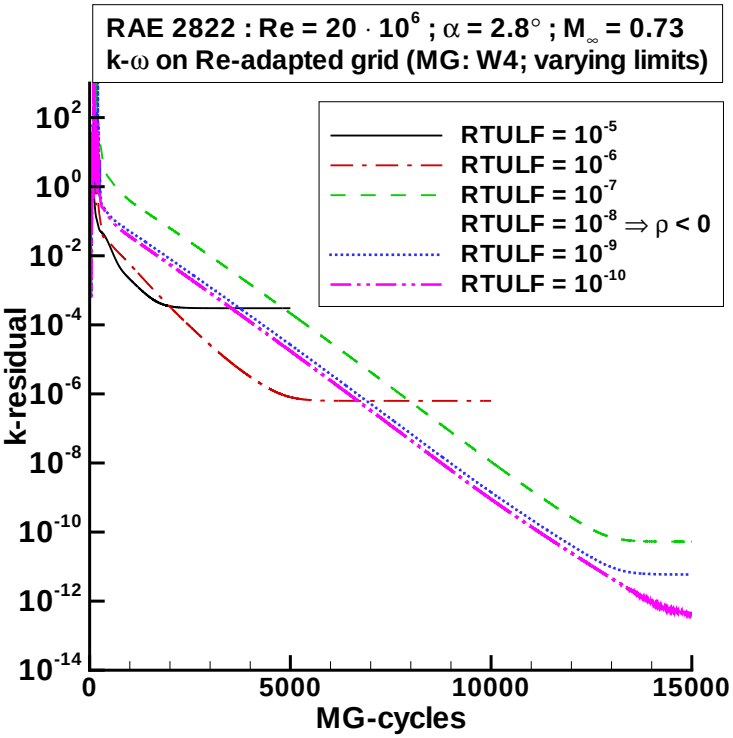

Figure 5.38: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 20 \cdot 10^6$; 4 multigrid level, W–cycle, variation of limiting factor of turbulence variables

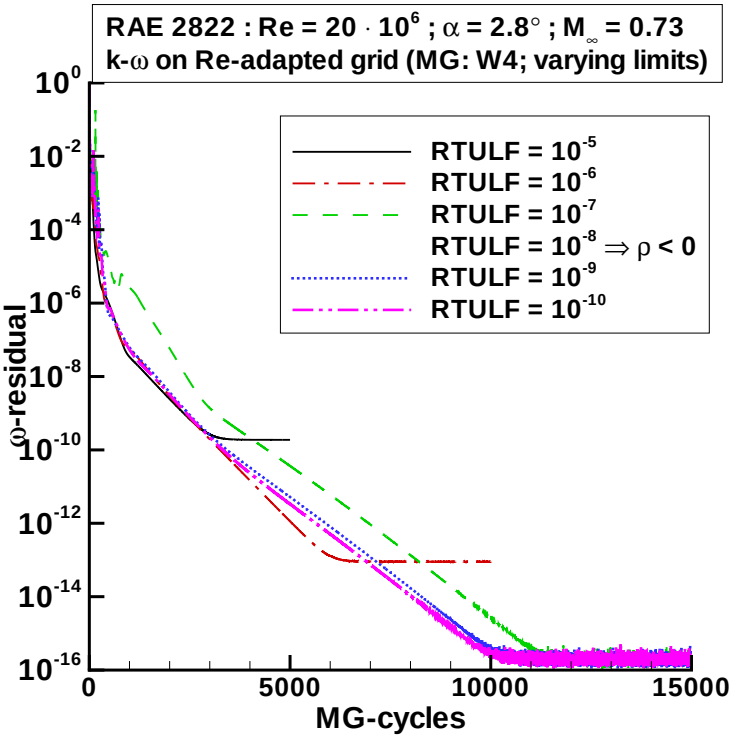

Figure 5.39: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, $\mathrm{Re} = 20 \cdot 10^6$; 4 multigrid level, W–cycle, variation of limiting factor of turbulence variables

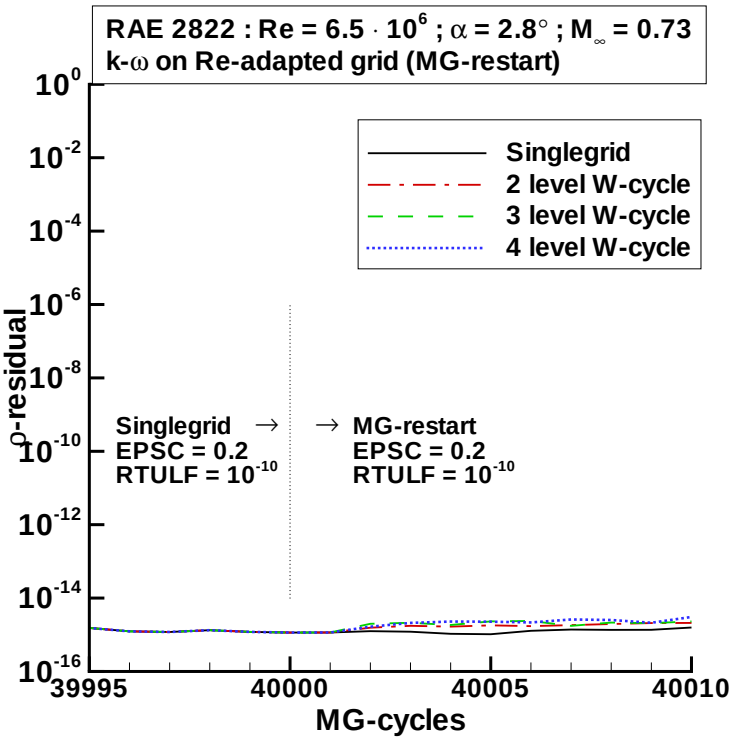

Figure 5.40: Close–up of convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, $\mathrm{Re} = 6.5 \cdot 10^6$, restart from singlegrid solution (obtained using $\mathtt{EPSC} = 0.2$, $\mathtt{RTULF} = 10^{-10}$) with varying number of multigrid levels in use

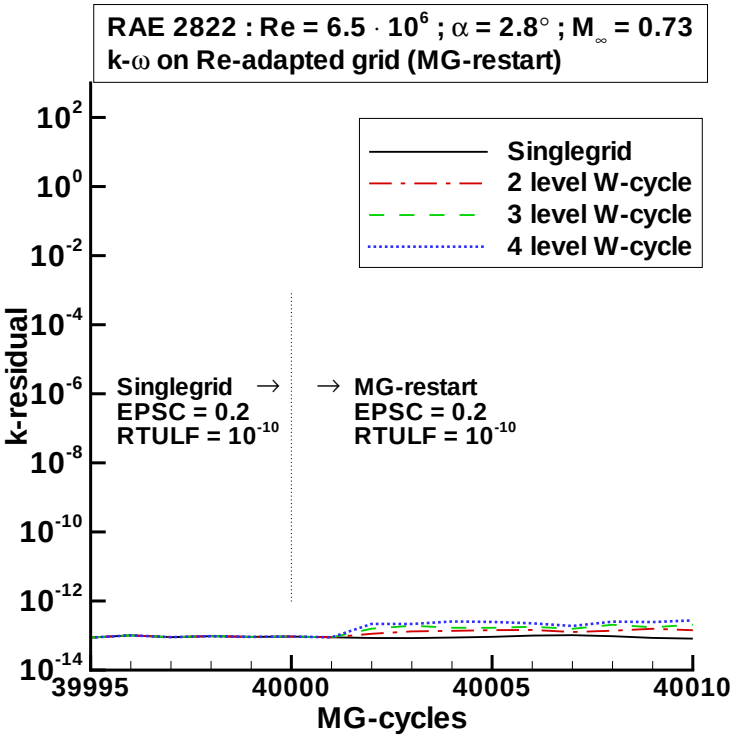

Figure 5.41: Close–up of convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 6.5 \cdot 10^6$, restart from singlegrid solution (obtained using EPSC $= 0.2$, RTULF $= 10^{-10}$) with varying number of multigrid levels in use

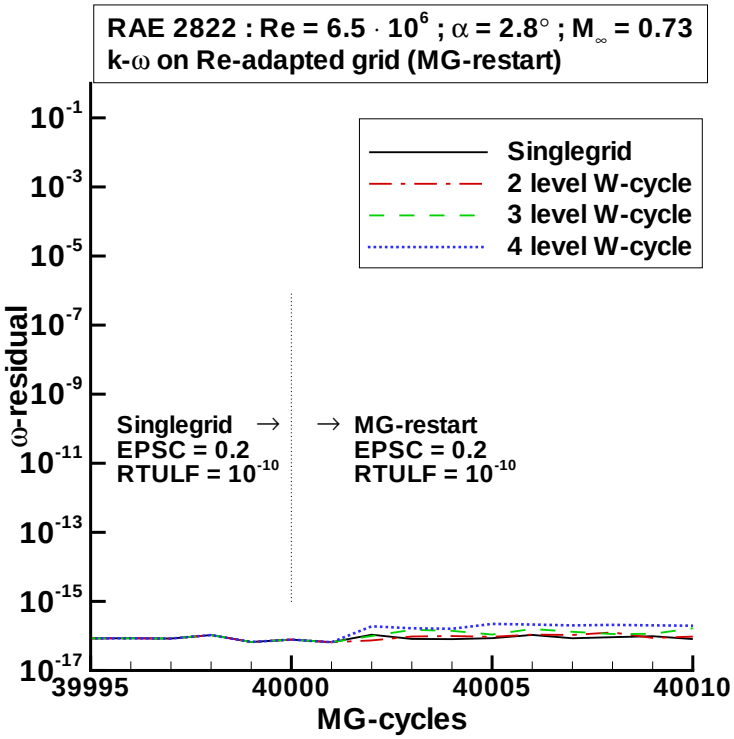

Figure 5.42: Close–up of convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, Re $= 6.5 \cdot 10^6$, restart from singlegrid solution (obtained using EPSC $= 0.2$, RTULF $= 10^{-10}$) with varying number of multigrid levels in use

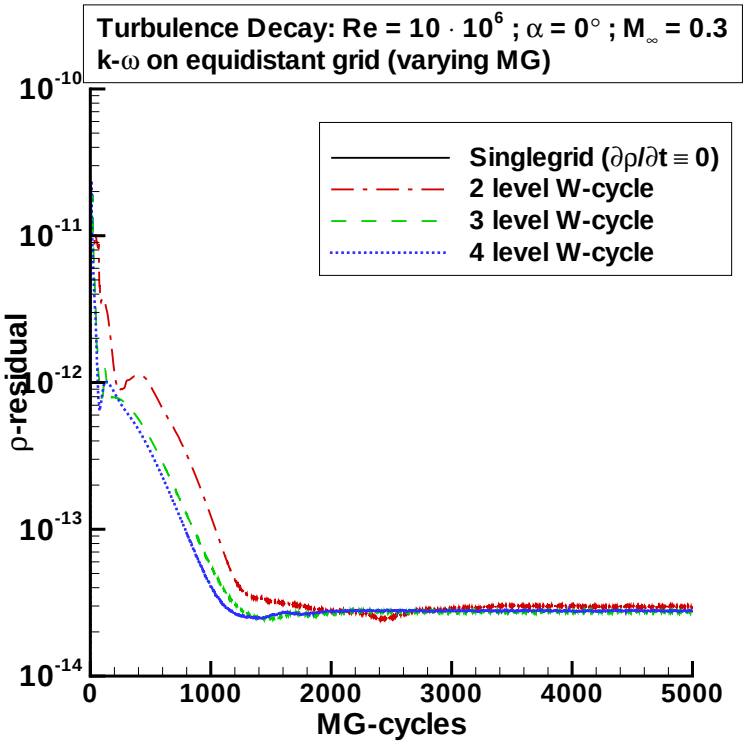

Figure 5.43: Convergence of density residual; turbulence decay, RANS calculation using k–ω turbulence model, varying number of multigrid levels in use, W–cycle

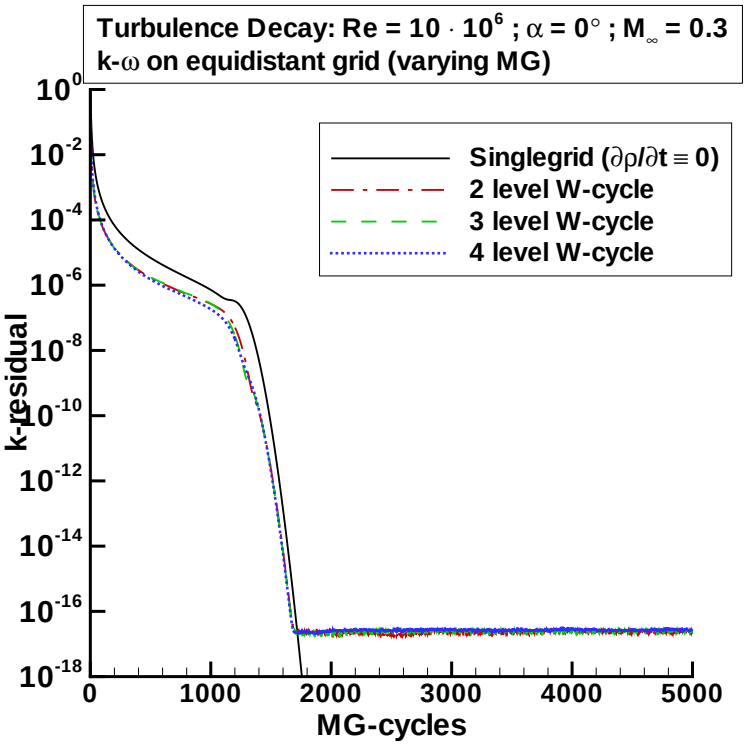

Figure 5.44: Convergence of k residual; turbulence decay, RANS calculation using k–ω turbulence model, varying number of multigrid levels in use, W–cycle

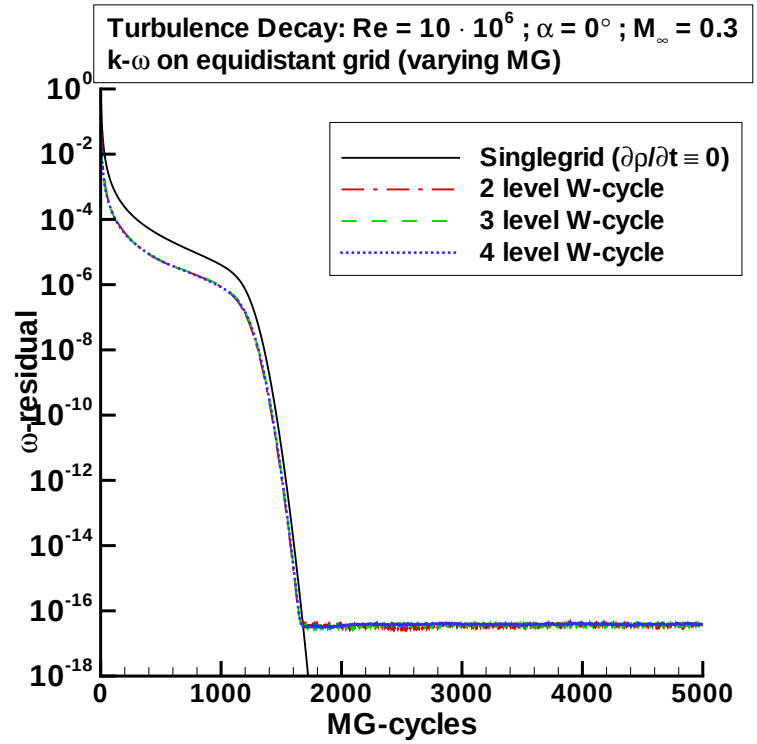

Figure 5.45: Convergence of ω residual; turbulence decay, RANS calculation using k–ω turbulence model, varying number of multigrid levels in use, W–cycle

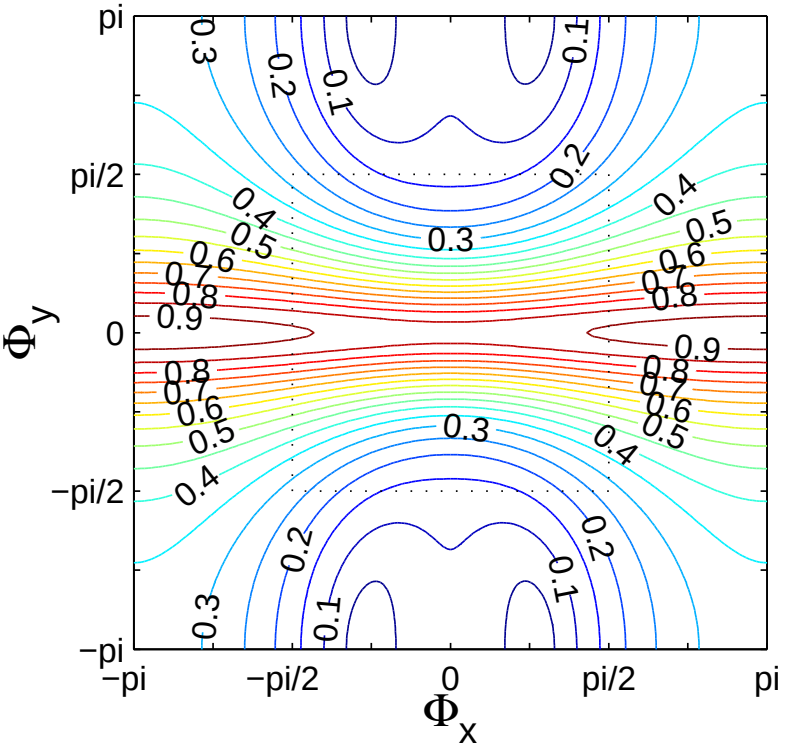

Figure 5.46: Fourier footprint of $g^{(q=5)}_{\text{pi-src, IRS}}$ with $\alpha_{\texttt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = -1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$

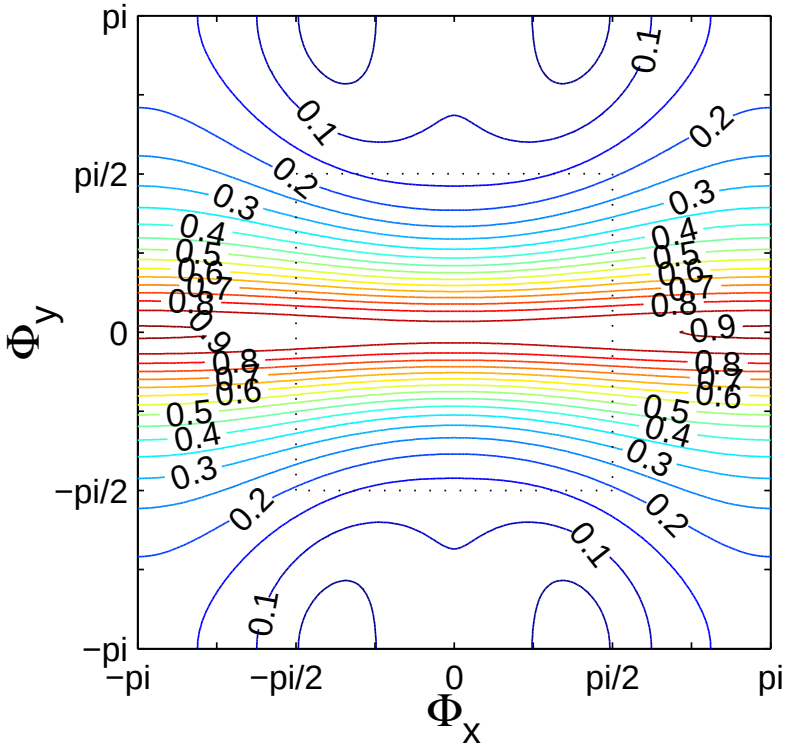

Figure 5.47: Fourier footprint of $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\texttt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = -1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; reduced implicit residual smoothing in x–direction

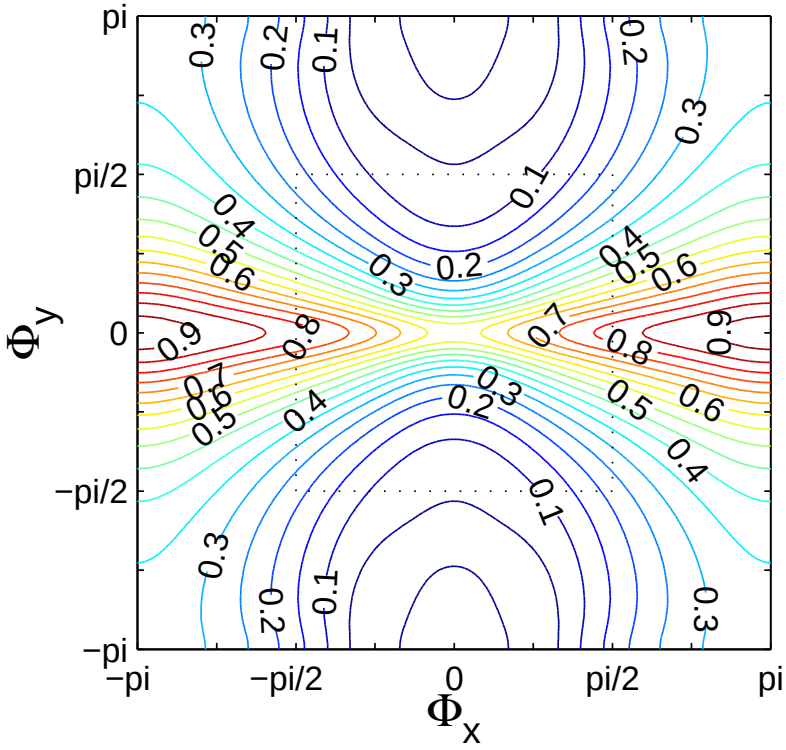

Figure 5.48: Fourier footprint of $g^{(q=5),\,2\mathrm{G}}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\texttt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = -1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 2 multigrid level ($\mathcal{P}^H_{2H} = 1\ \forall H$)

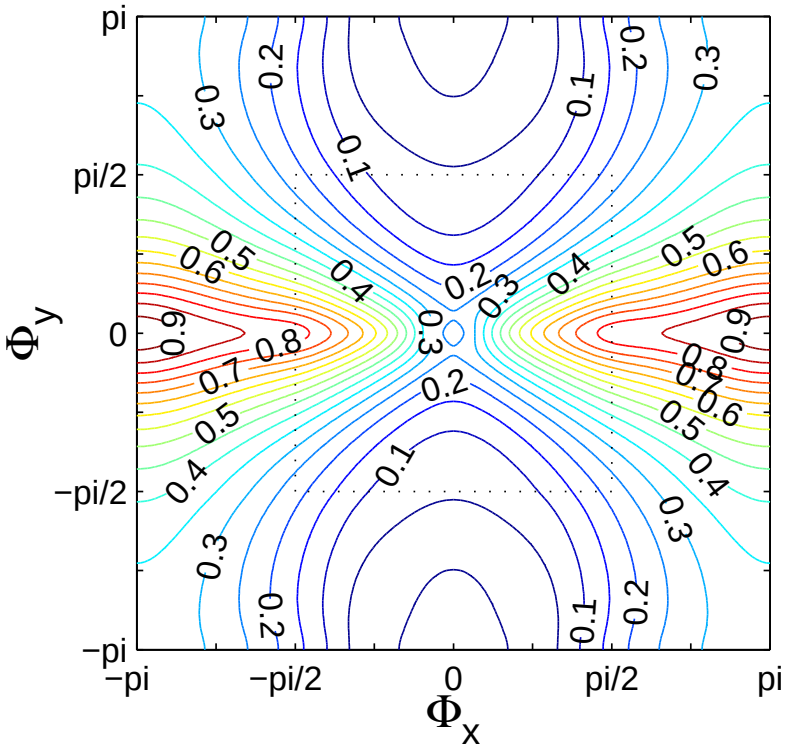

Figure 5.49: Fourier footprint of $g^{(q=5),\,\mathrm{MG}}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = -1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 3 multigrid level ($\mathcal{P}^H_{2H} = 1\ \forall H$)

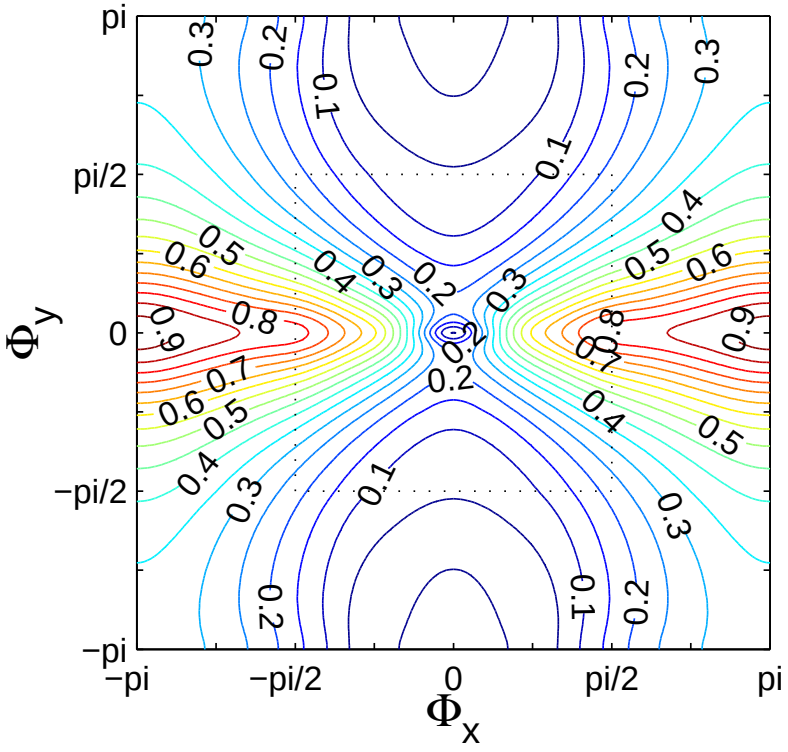

Figure 5.50: Fourier footprint of $g^{(q=5),\,\mathrm{MG}}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = -1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 4 multigrid level ($\mathcal{P}^H_{2H} = 1\ \forall H$)

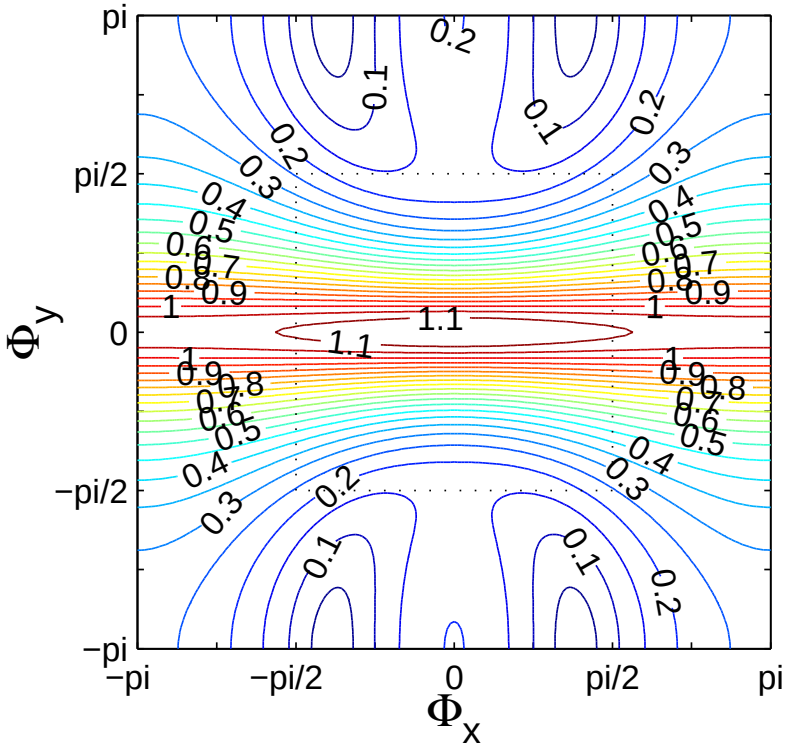

Figure 5.51: Fourier footprint of $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$

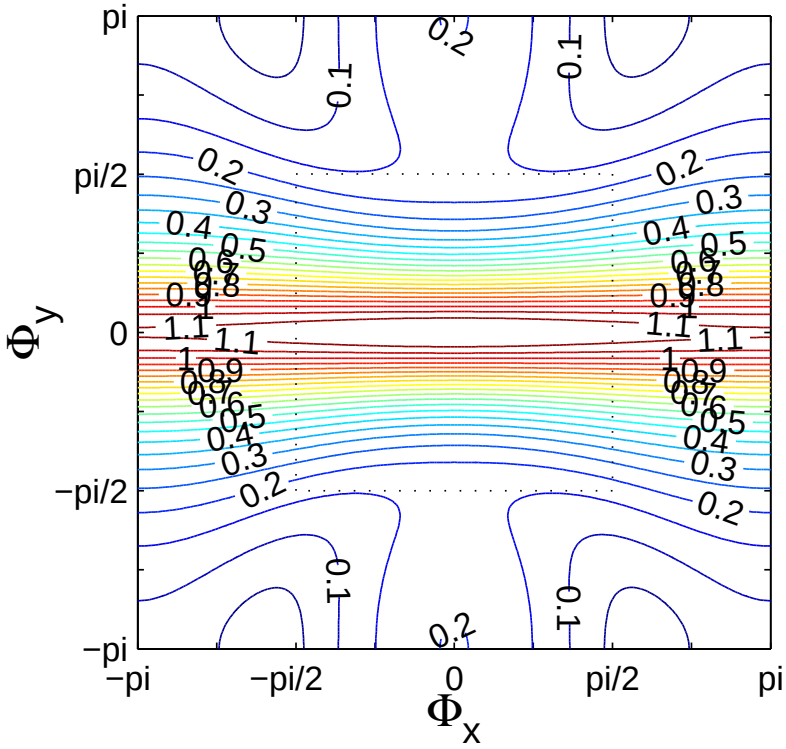

Figure 5.52: Fourier footprint of $g^{(q=5)}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; reduced implicit residual smoothing in x–direction

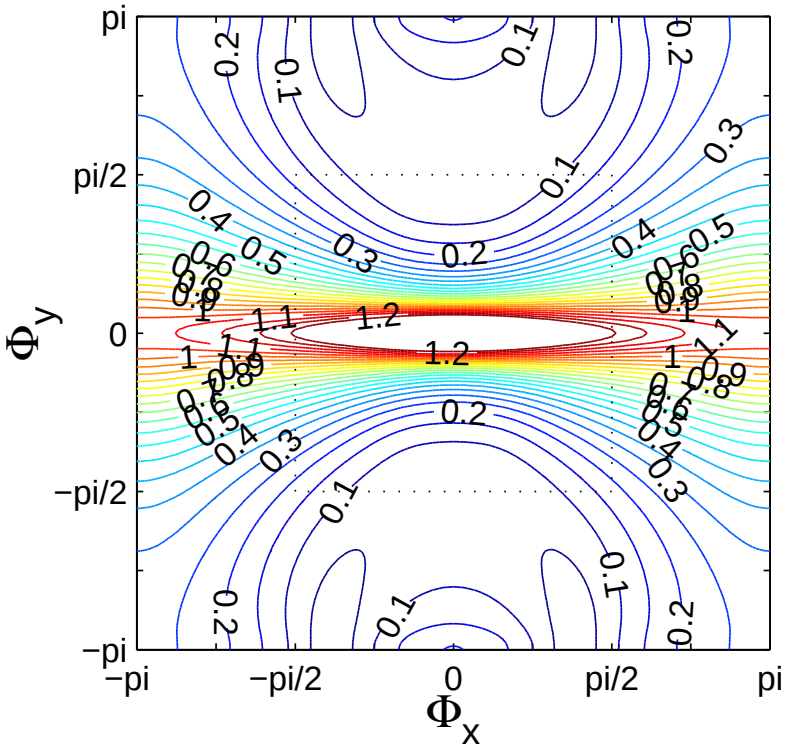

Figure 5.53: Fourier footprint of $g^{(q=5),\,2\mathrm{G}}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 2 multigrid level ($\mathcal{P}^H_{2H} = 1\ \forall H$)

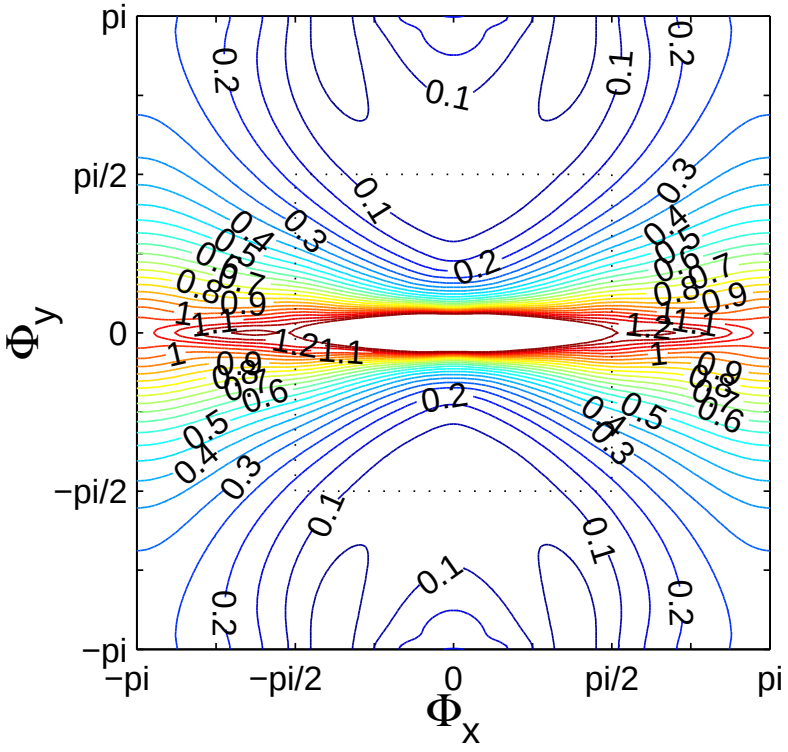

Figure 5.54: Fourier footprint of $g^{(q=5),\,\mathrm{MG}}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 3 multigrid level ($\mathcal{P}^H_{2H} = 1\ \forall H$)

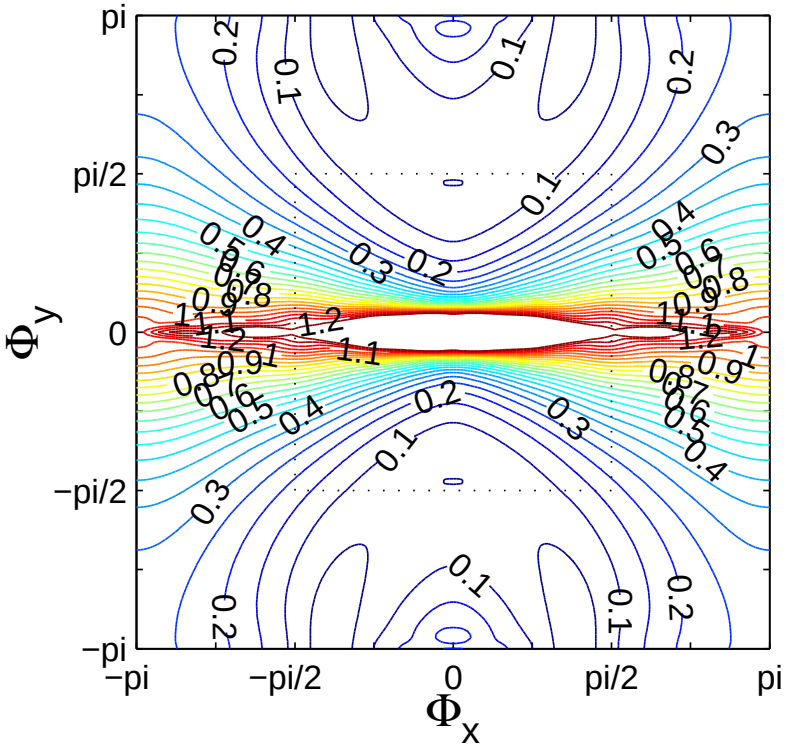

Figure 5.55: Fourier footprint of $g^{(q=5),\,\mathrm{MG}}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\texttt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 4 multigrid level ($\mathcal{P}^H_{2H} = 1\ \forall H$)

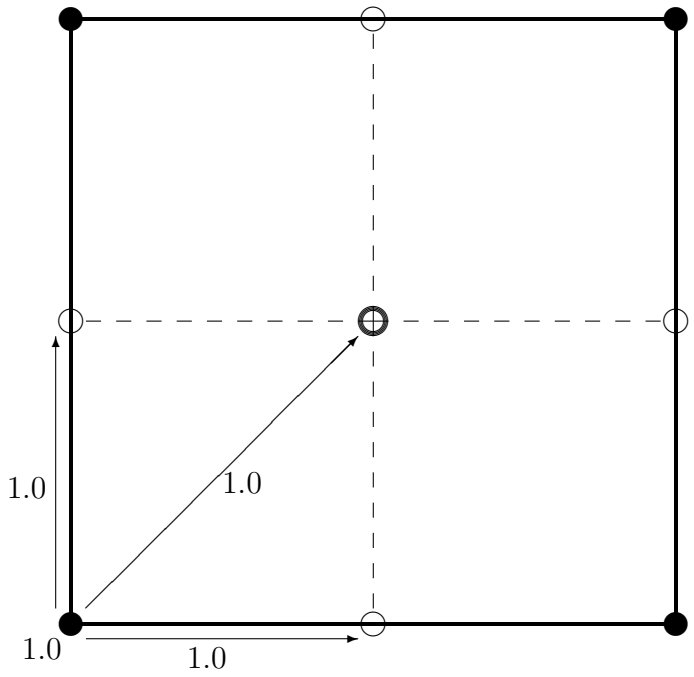

Figure 6.1: Schematic description of upwind prolongation (2D), i. e. Eqs. (3.111) and (3.112) using $a^- = b^- = 1.0$, $a^+ = b^+ = 0.0$ and $c = 1.0$

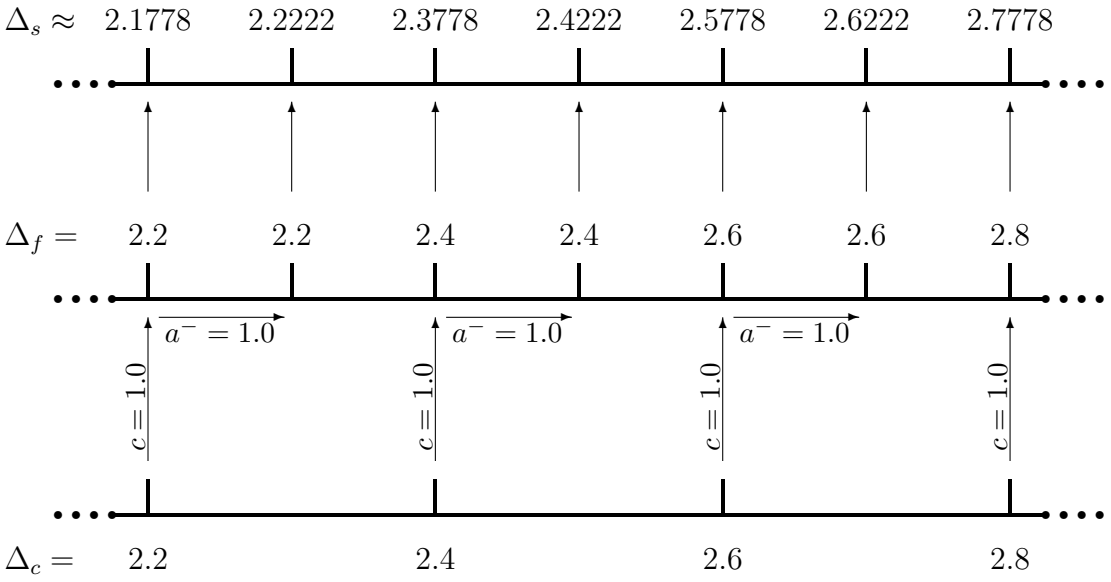

Figure 6.2: Effect of upwind prolongation (1D), i. e. Eq. (3.111) using $a^- = 1.0$, $a^+ = 0.0$ and $c = 1.0$

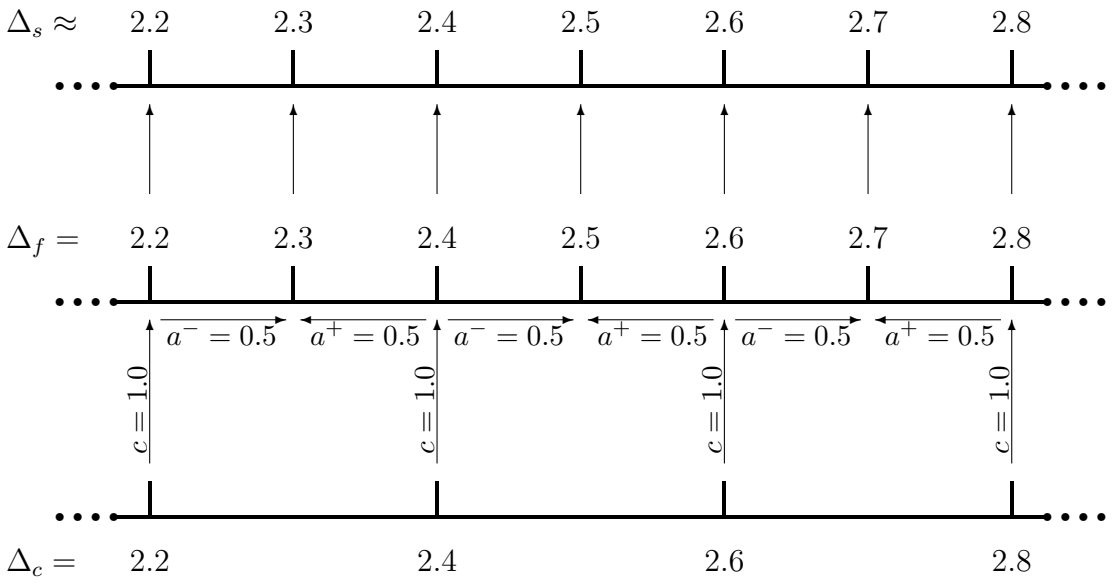

Figure 6.3: Effect of linear interpolation (1D), i. e. Eq. (3.111) using $a^- = 0.5$, $a^+ = 0.5$ and $c = 1.0$

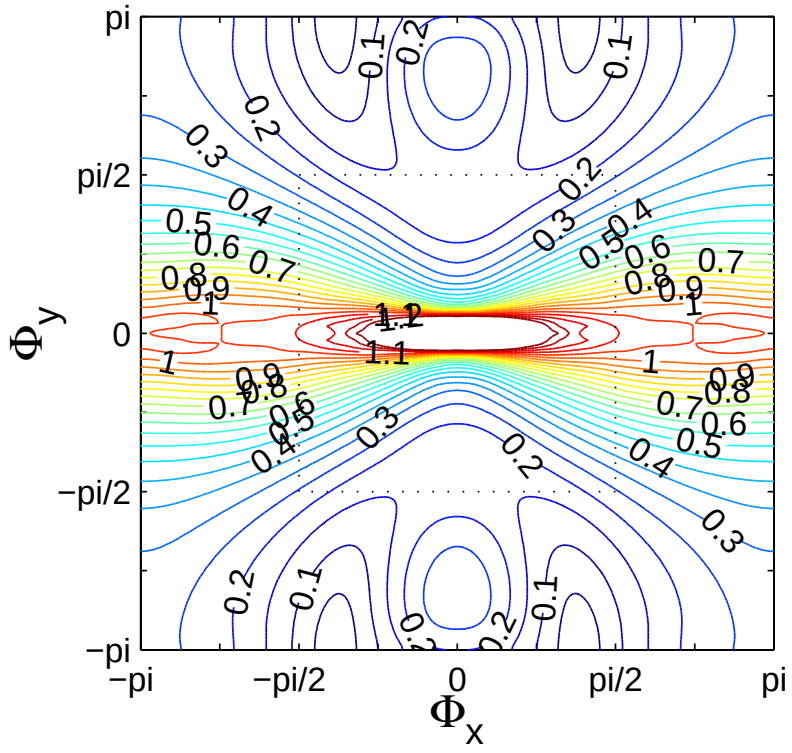

Figure 6.4: Fourier footprint of $g^{(q=5),\,\mathrm{MG}}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 4 multigrid level with bilinear interpolation prolongation ($\mathcal{P}^H_{2H} = \mathcal{P}^H_{[2]H}\big|_j \mathcal{P}^H_{[2]H}\big|_i \widetilde{\mathcal{P}}^H_{2H}\ \forall H$ with $a^- = a^+ = b^- = b^+ = \frac{1}{2}$ and $c = 1.0$)

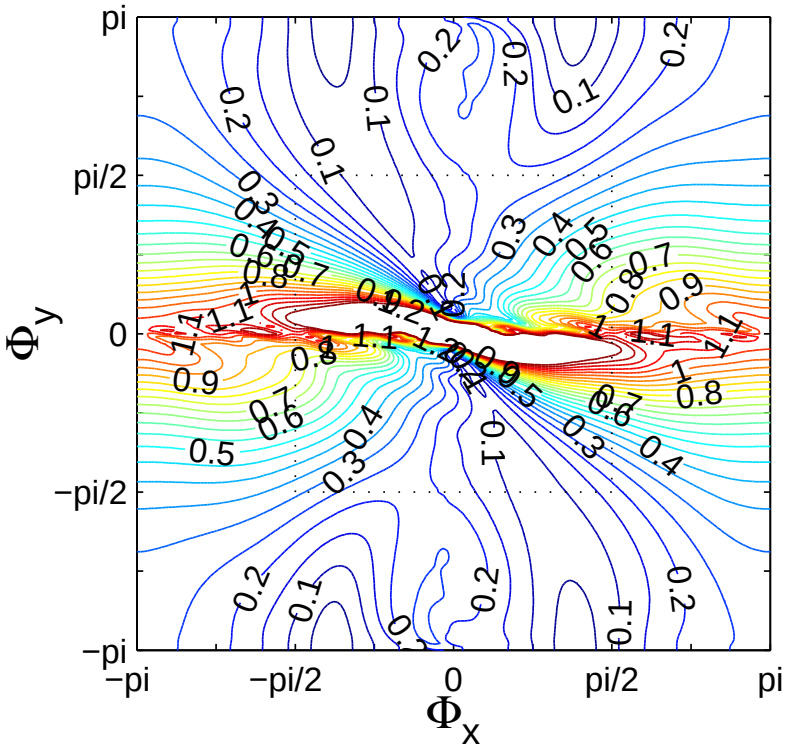

Figure 6.5: Fourier footprint of $g^{(q=5),\,\mathrm{MG}}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 4 multigrid level with upwind directed prolongation ($\mathcal{P}^H_{2H} = \mathcal{P}^H_{[2]H}\big|_j\, \mathcal{P}^H_{[2]H}\big|_i\, \widetilde{\mathcal{P}}^H_{2H}\ \forall H$ with $a^- = b^- = 1.0$, $a^+ = b^+ = 0.0$ and $c = 1.0$)

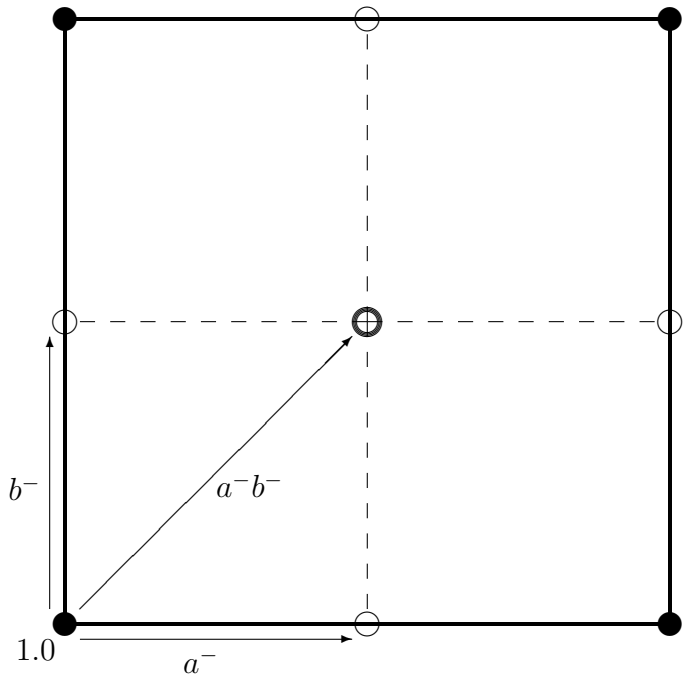

Figure 6.6: Schematic description of upwind prolongation (2D), i. e. Eqs. (3.111) and (3.112) using $0 \le a^-, b^- \le 1.0$, $a^+ = b^+ = 0.0$ and $c = 1.0$

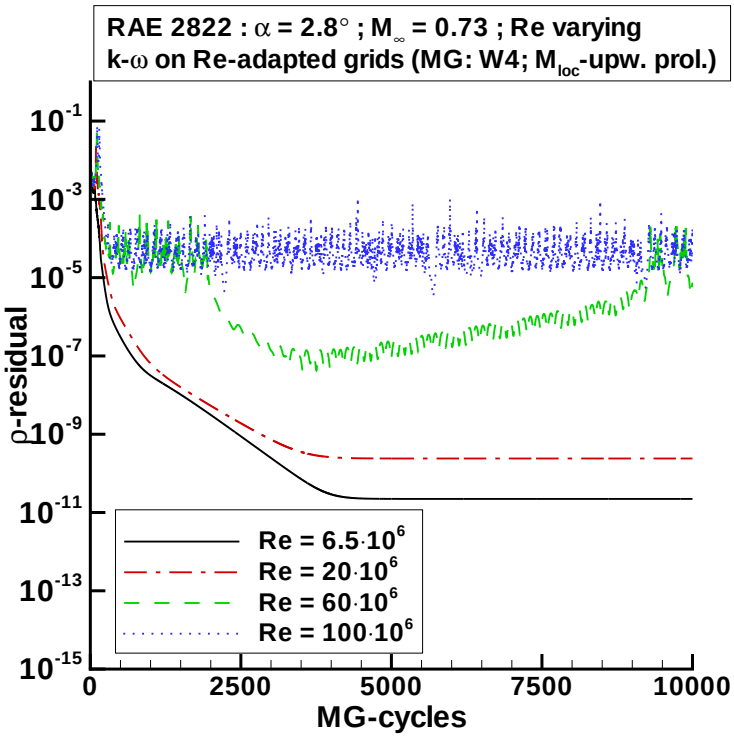

Figure 6.7: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, varying Reynolds number; 4 multigrid level, W-cycle k–ω: weighted upwind prolongation operator

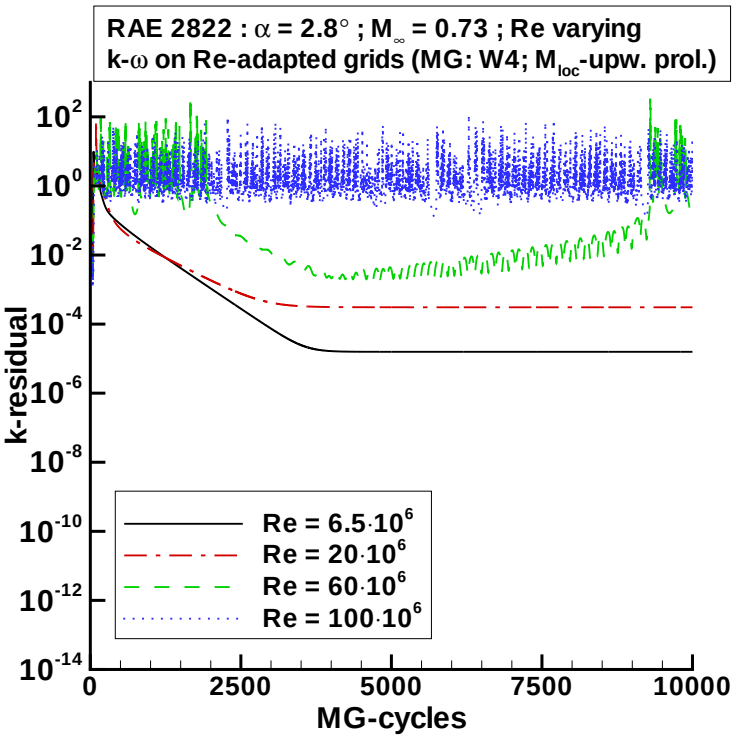

Figure 6.8: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, varying Reynolds number; 4 multigrid level, W-cycle k–ω: weighted upwind prolongation operator

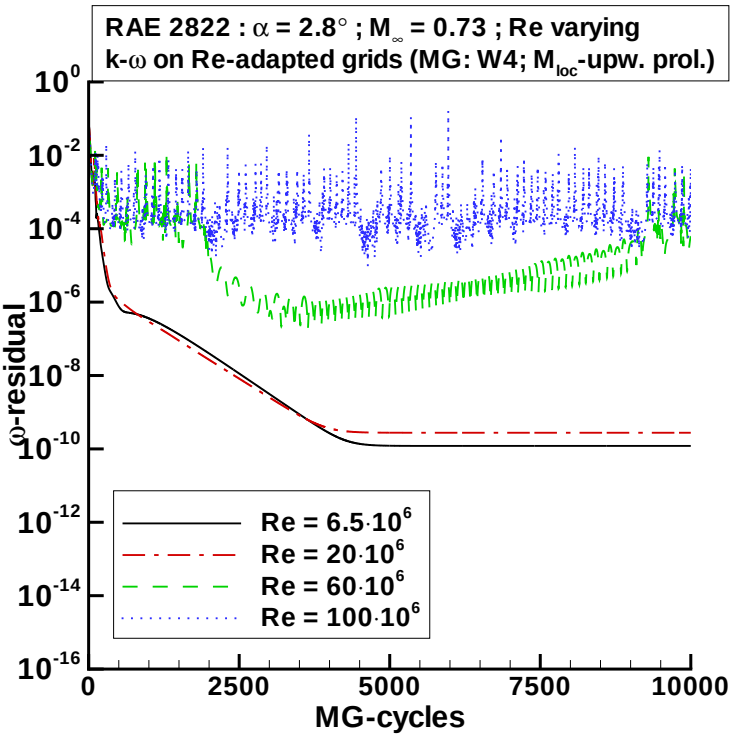

Figure 6.9: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, varying Reynolds number; 4 multigrid level, W-cycle k–ω: weighted upwind prolongation operator

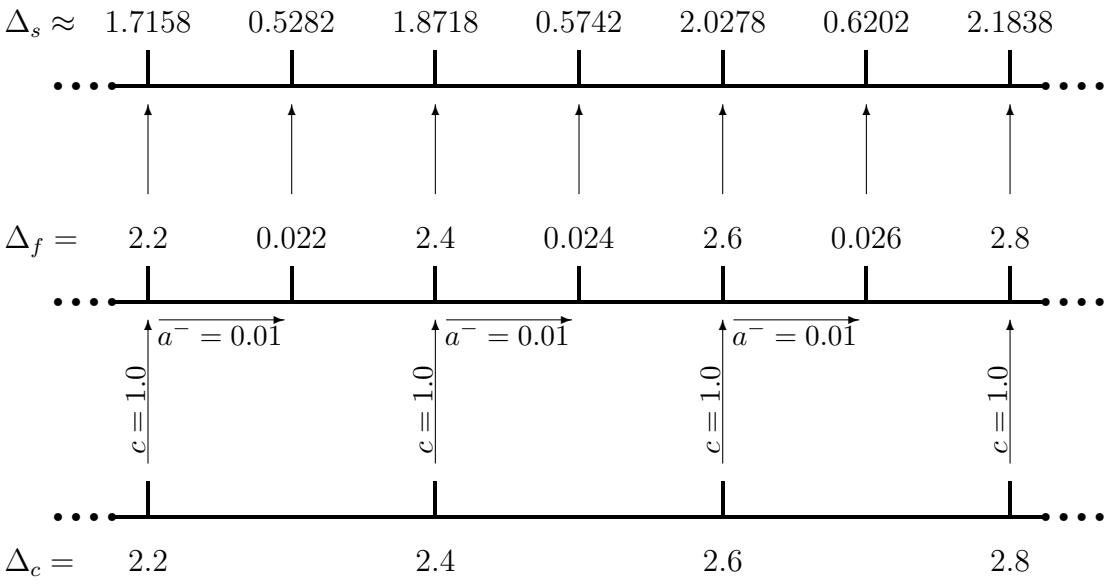

Figure 6.10: Effect of weighted upwind prolongation (1D), i. e. Eq. (3.111) using $a^- = 0.01$, $a^+ = 0.0$ and $c = 1.0$

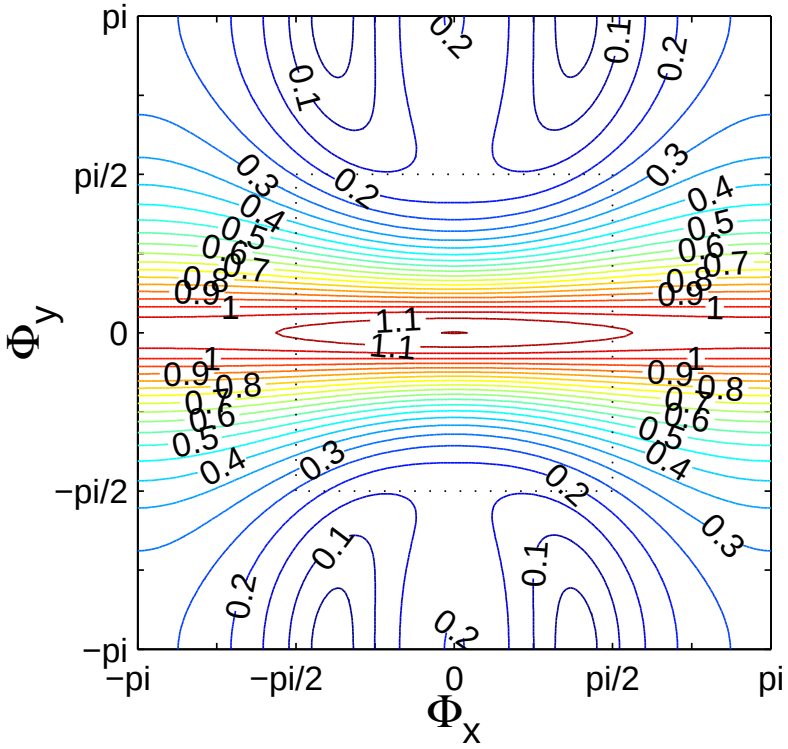

Figure 6.11: Fourier footprint of $g^{(q=5),\,\mathrm{MG}}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 4 multigrid level with upwind weighted prolongation using velocity as weighting factor ($\mathcal{P}^H_{2H} = \mathcal{P}^H_{[2]H}\big|_j\, \mathcal{P}^H_{[2]H}\big|_i\, \widetilde{\mathcal{P}}^H_{2H}\ \forall H$ with $a^- = u$, $b^- = v$, $a^+ = b^+ = 0.0$ and $c = 1.0$)

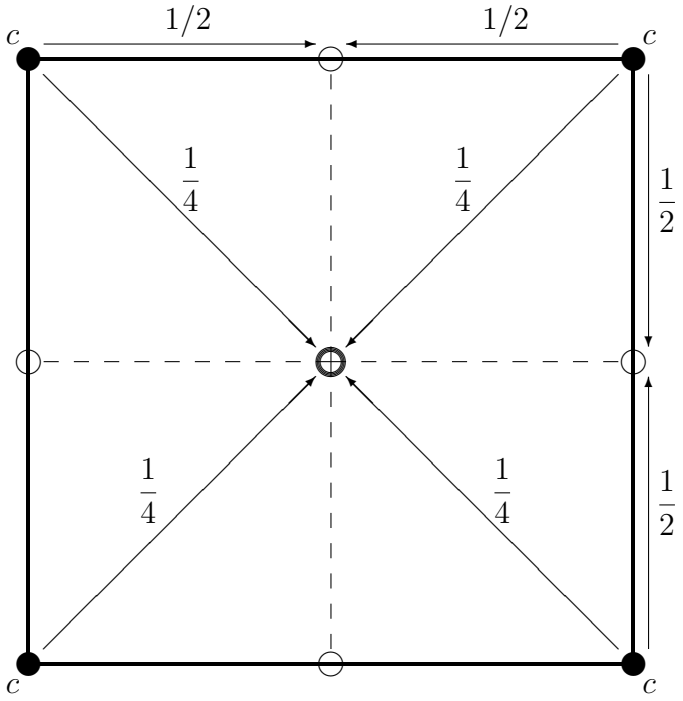

Figure 6.12: Schematic description of relaxed bilinear interpolation (2D), i. e. Eqs. (3.111) and (3.112) using $a^- = a^+ = b^- = b^+ = 0.5$ and $0.0 \le c \le 1.0$

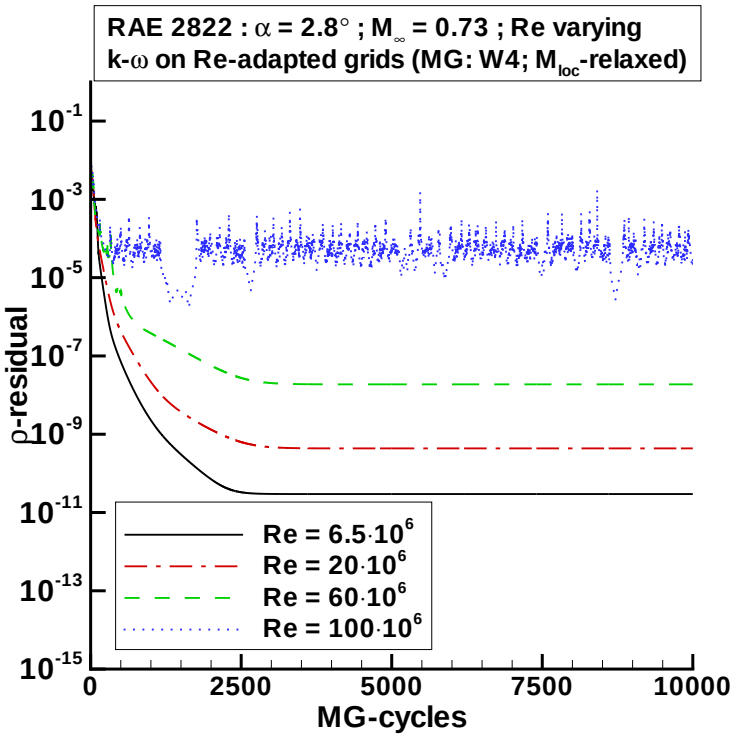

Figure 6.13: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, varying Reynolds number; 4 multigrid level, W-cycle k–ω: relaxed linear prolongation operator

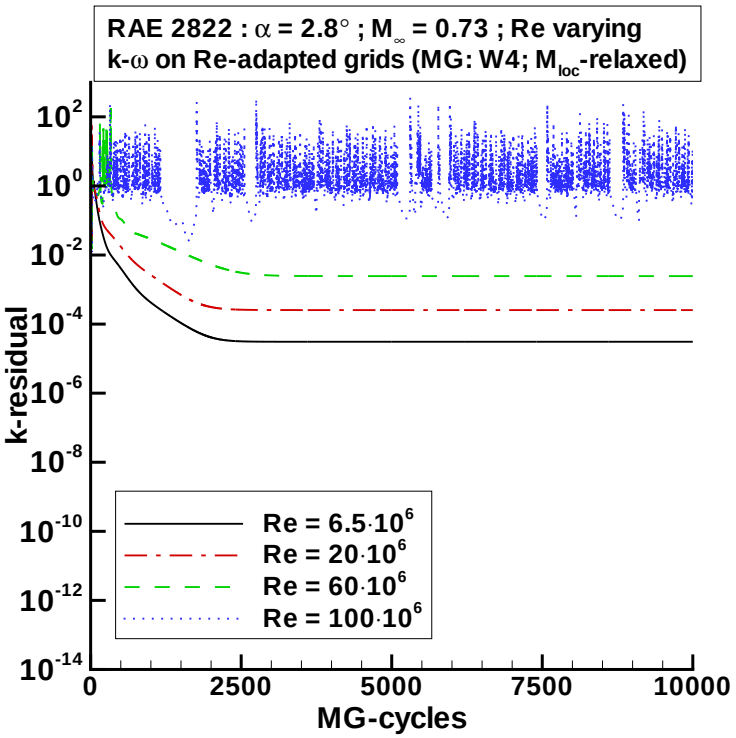

Figure 6.14: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, varying Reynolds number; 4 multigrid level, W-cycle k–ω: relaxed linear prolongation operator

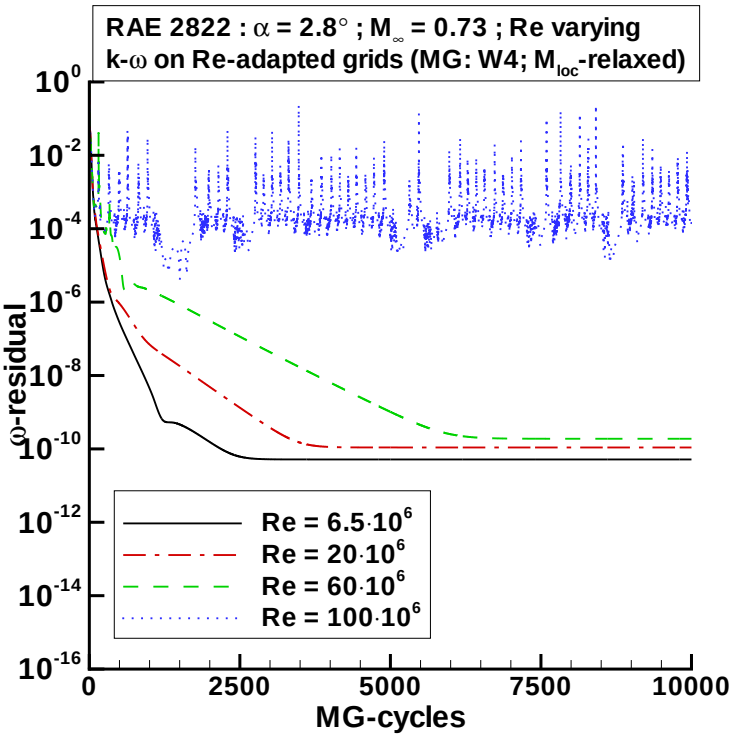

Figure 6.15: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, varying Reynolds number; 4 multigrid level, W-cycle k–ω: relaxed linear prolongation operator

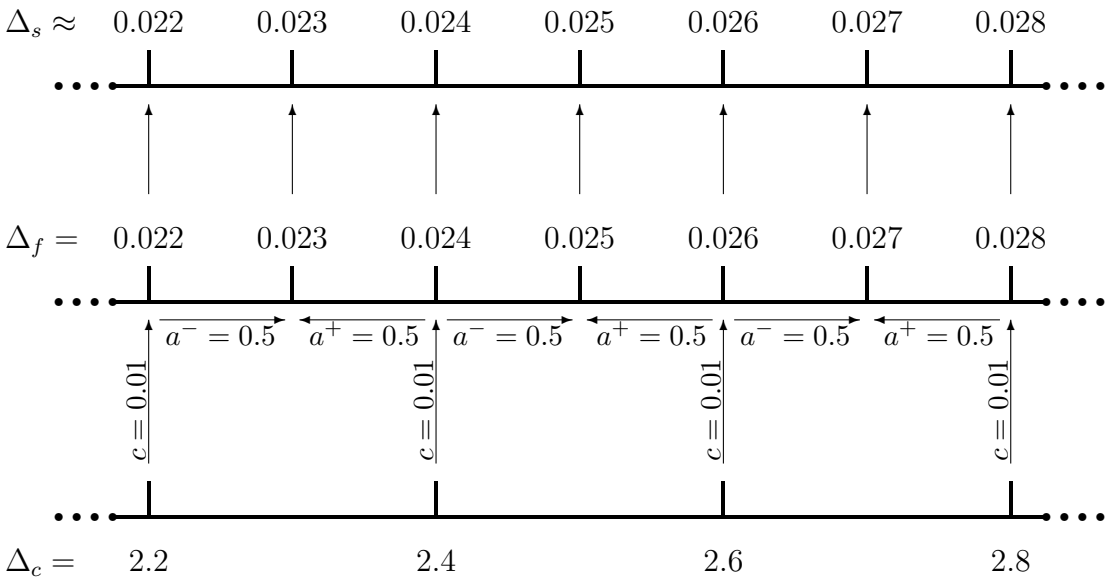

Figure 6.16: Effect of relaxed linear interpolation (1D), i. e. Eq. (3.111) using $a^- = 0.5$, $a^+ = 0.5$ and $c = 0.01$

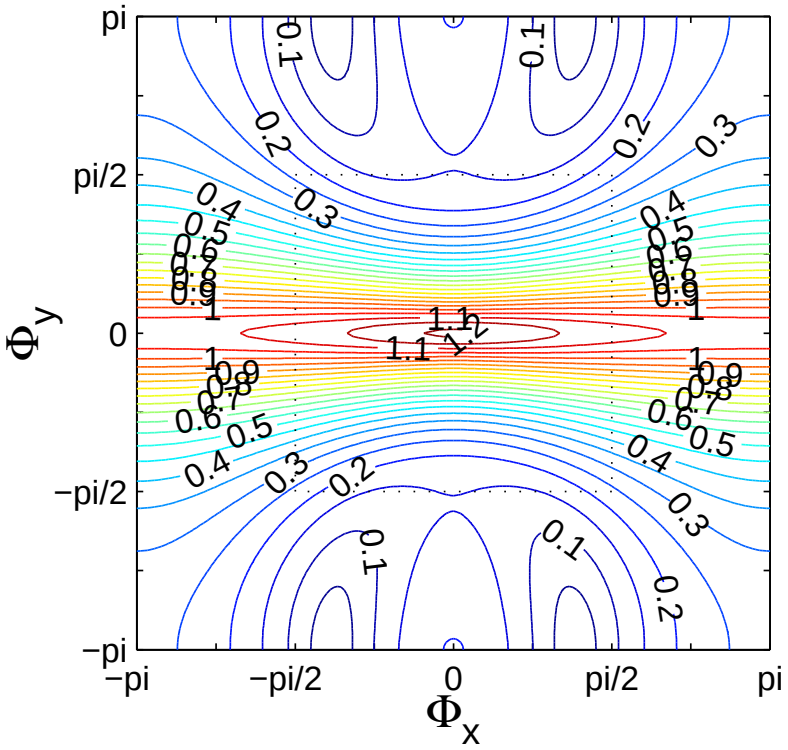

Figure 6.17: Fourier footprint of $g^{(q=5),\,2\mathrm{G}}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 2 multigrid level ($\mathcal{P}^H_{2H} = 0.1\ \forall H$)

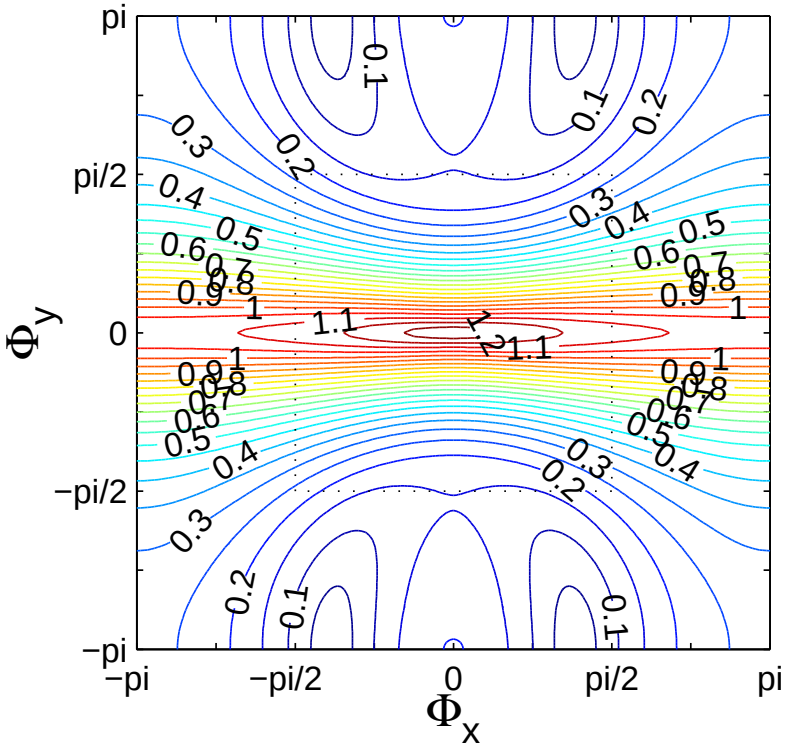

Figure 6.18: Fourier footprint of $g^{(q=5),\,\mathrm{MG}}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 3 multigrid level ($\mathcal{P}^H_{2H} = 0.1\ \forall H$)

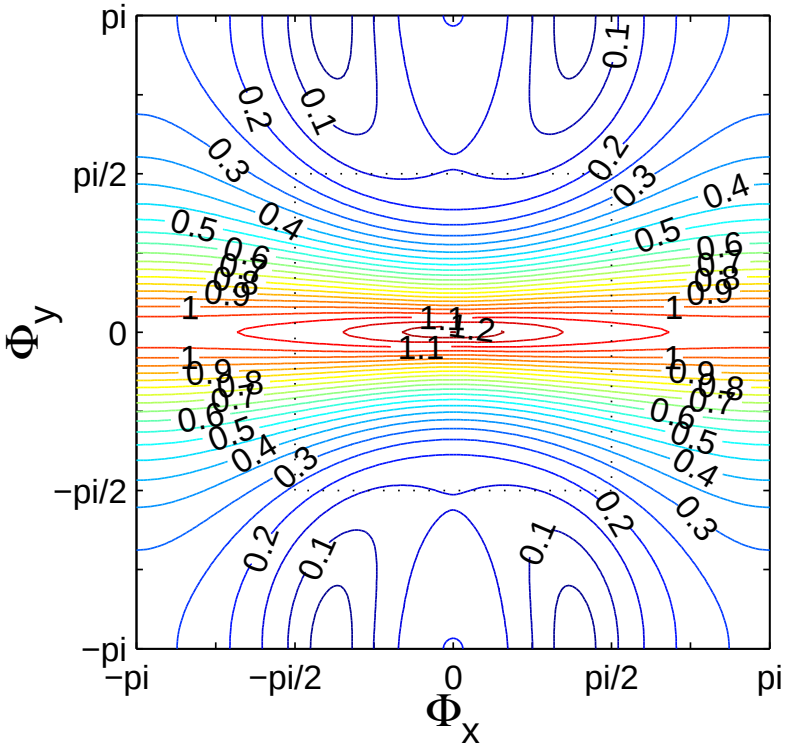

Figure 6.19: Fourier footprint of $g^{(q=5),\,\mathrm{MG}}_{\mathrm{pi-src,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 4 multigrid level ($\mathcal{P}^H_{2H} = 0.1\ \forall H$)

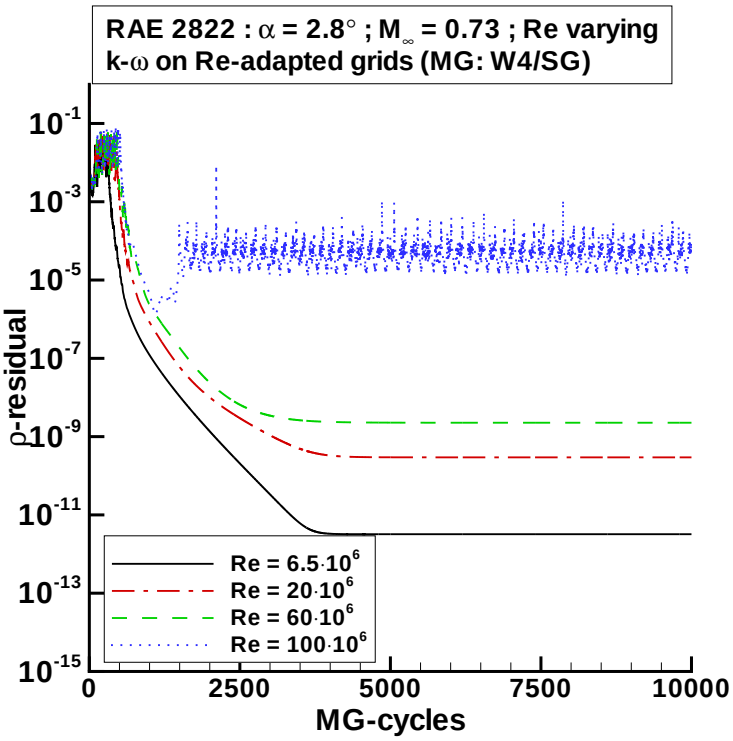

Figure 6.20: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: single-grid ($\mathrm{Re} = 6.5 \cdot 10^6$: $\mathtt{CFLTU} = 10$)

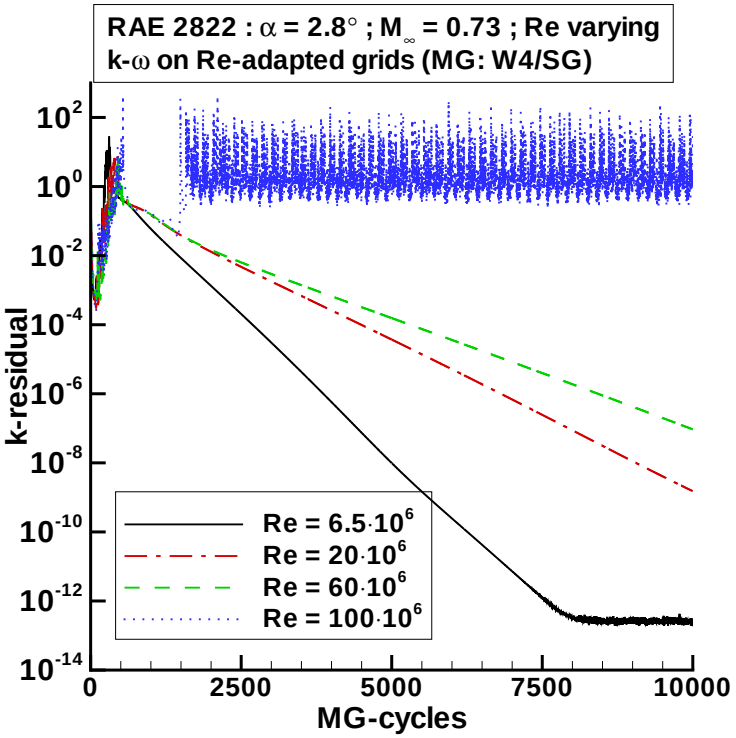

Figure 6.21: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid (Re $= 6.5 \cdot 10^6$: CFLTU $= 10$)

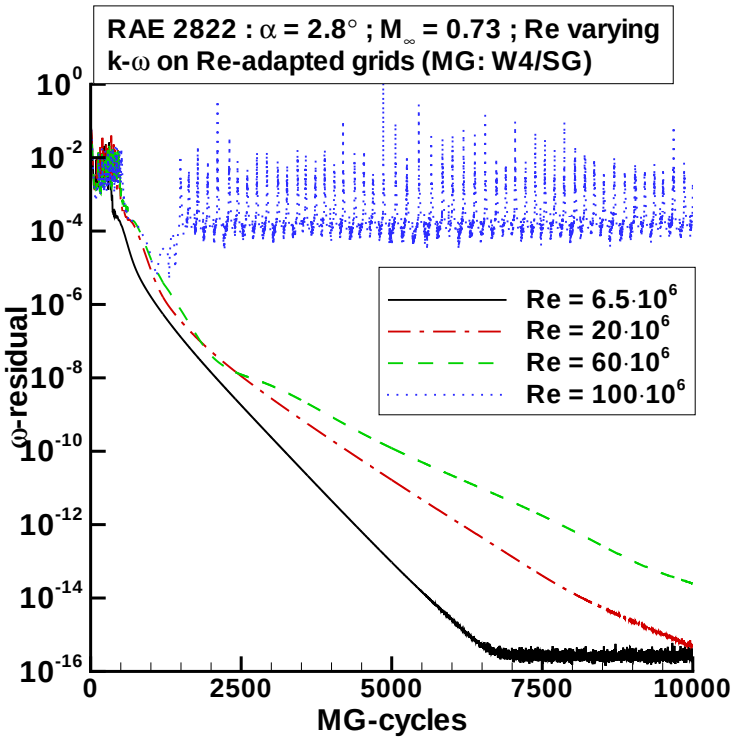

Figure 6.22: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid (Re $= 6.5 \cdot 10^6$: CFLTU $= 10$)

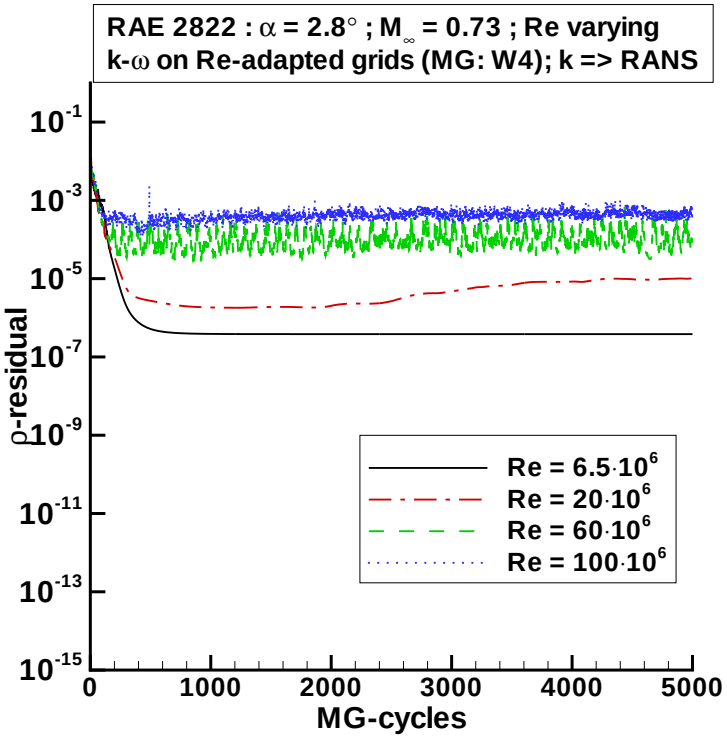

Figure 6.23: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with maximized coupling, varying Reynolds number; 4 multigrid level, W-cycle

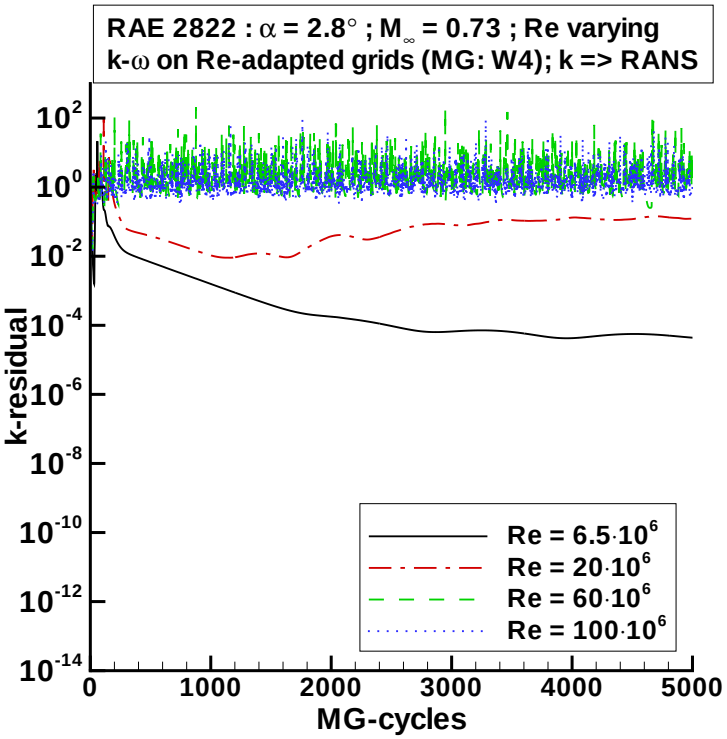

Figure 6.24: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with maximized coupling, varying Reynolds number; 4 multigrid level, W-cycle

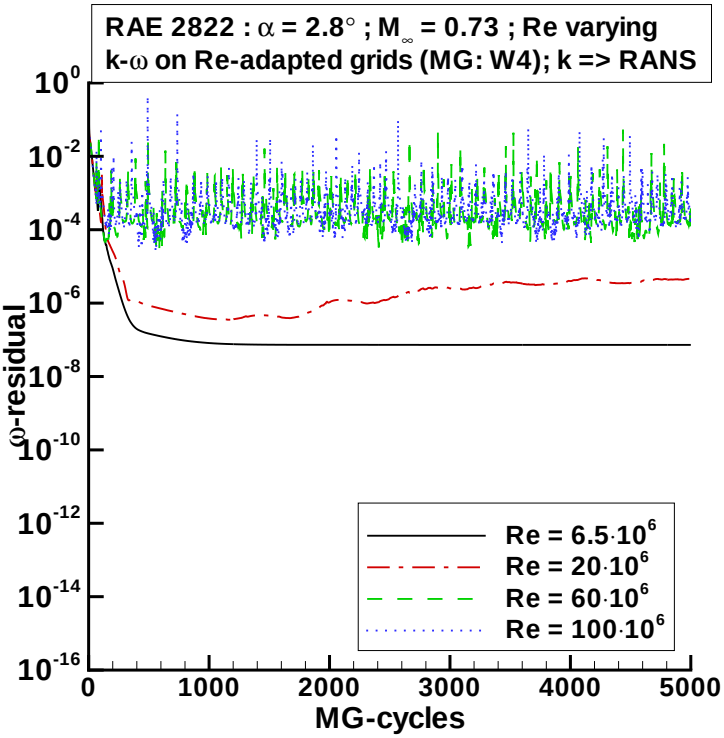

Figure 6.25: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with maximized coupling, varying Reynolds number; 4 multigrid level, W-cycle

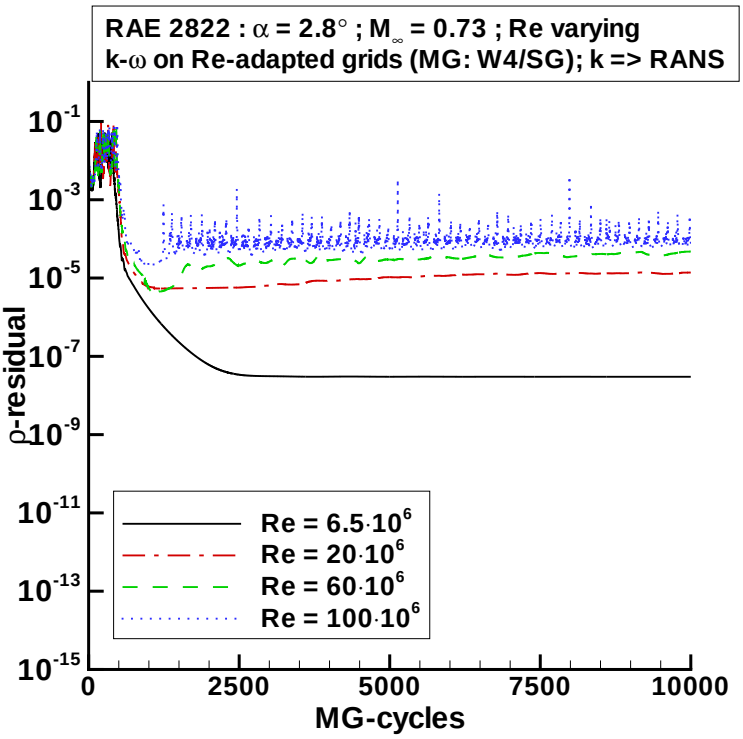

Figure 6.26: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with maximized coupling, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

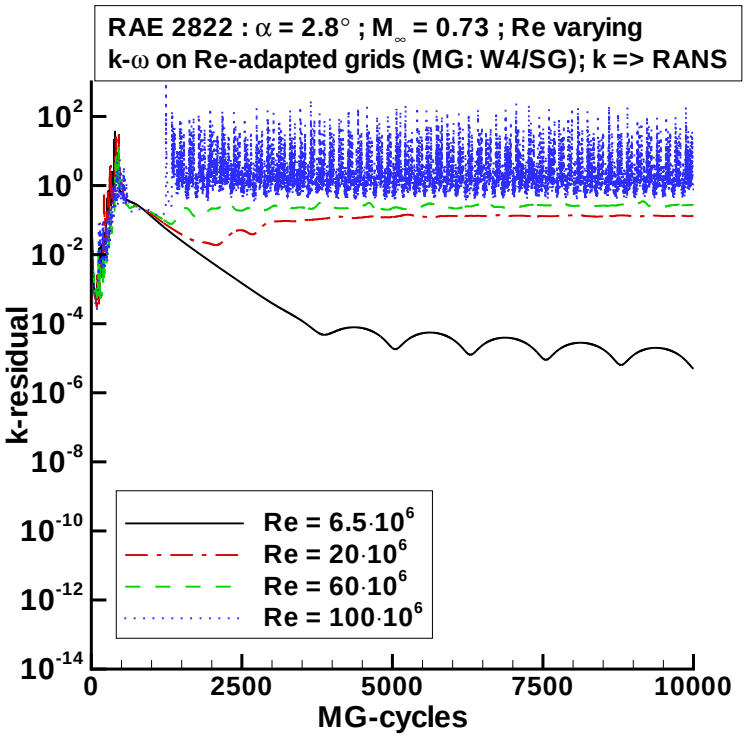

Figure 6.27: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with maximized coupling, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

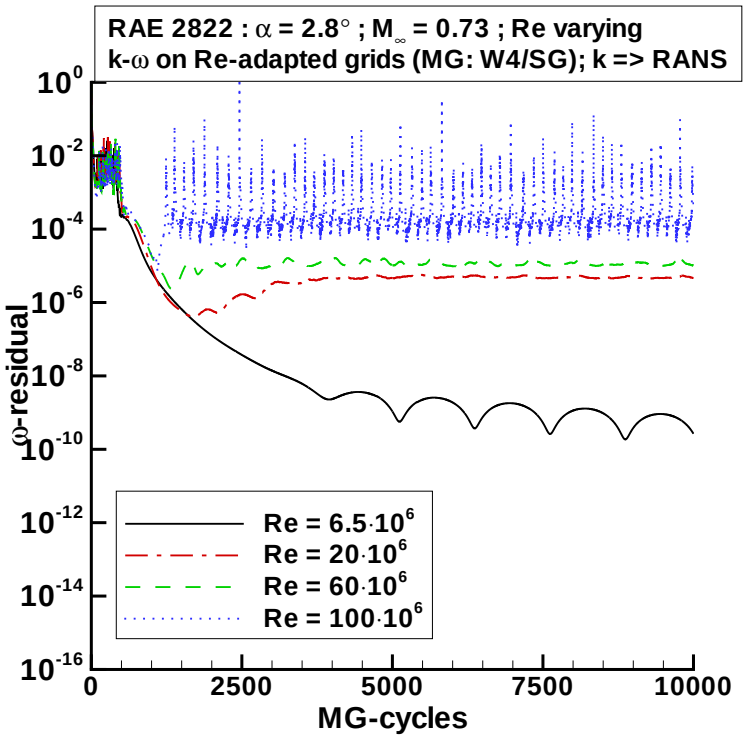

Figure 6.28: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with maximized coupling, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

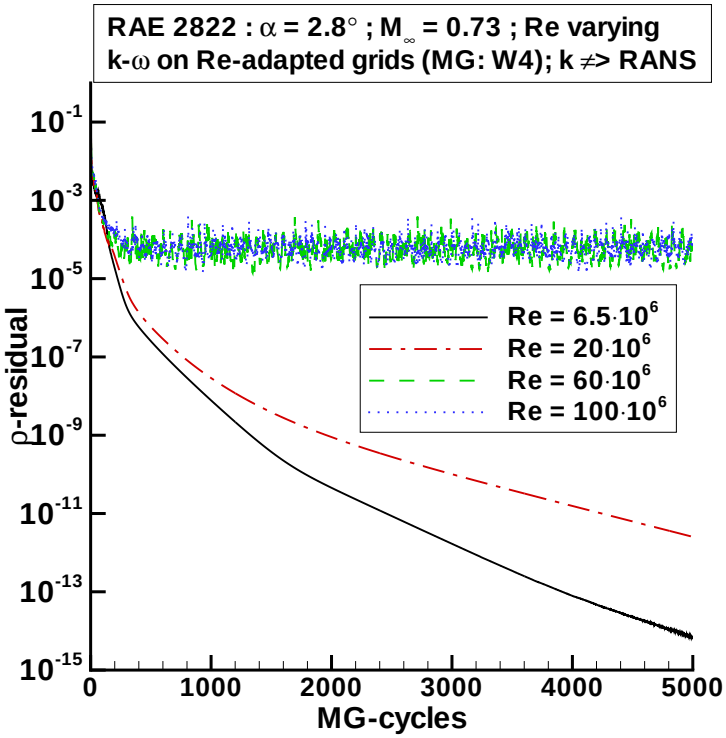

Figure 6.29: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with minimized coupling, varying Reynolds number; 4 multigrid level, W-cycle

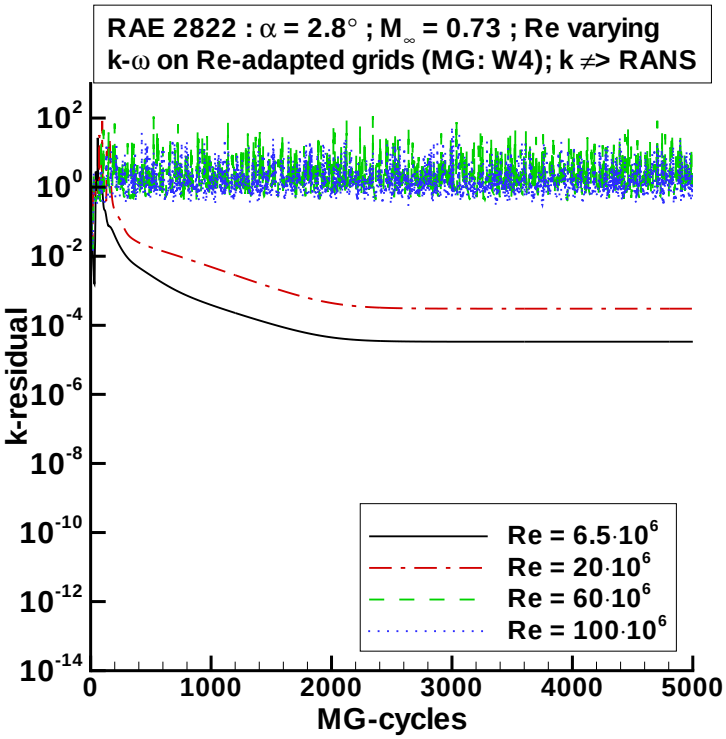

Figure 6.30: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with minimized coupling, varying Reynolds number; 4 multigrid level, W-cycle

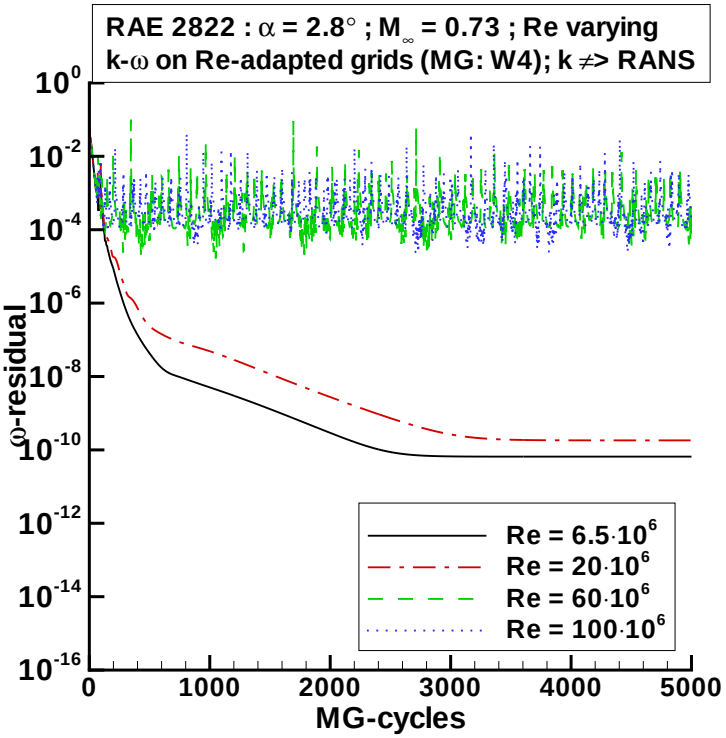

Figure 6.31: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with minimized coupling, varying Reynolds number; 4 multigrid level, W-cycle

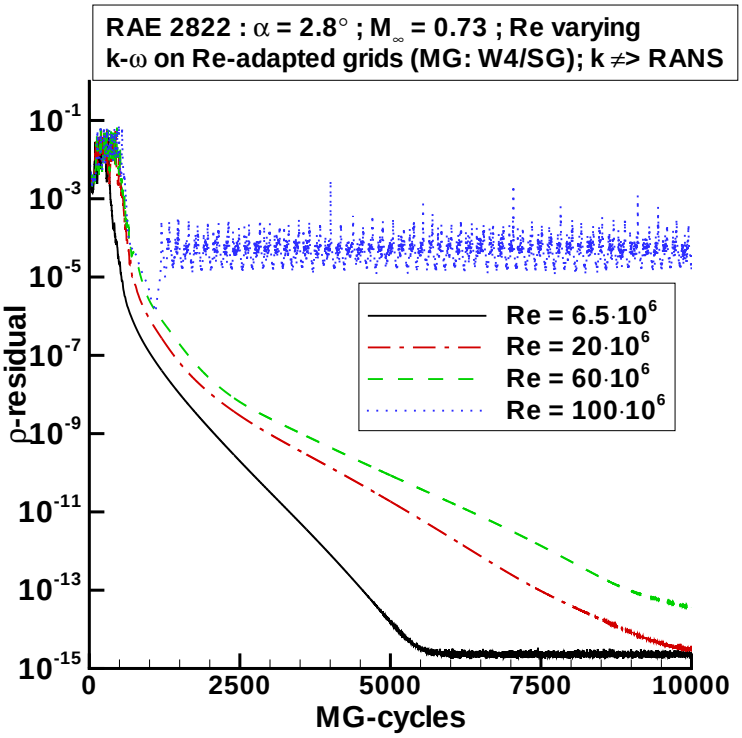

Figure 6.32: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with minimized coupling, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid ($\mathrm{Re} = 6.5 \cdot 10^6$: `CFLTU` $= 10$)

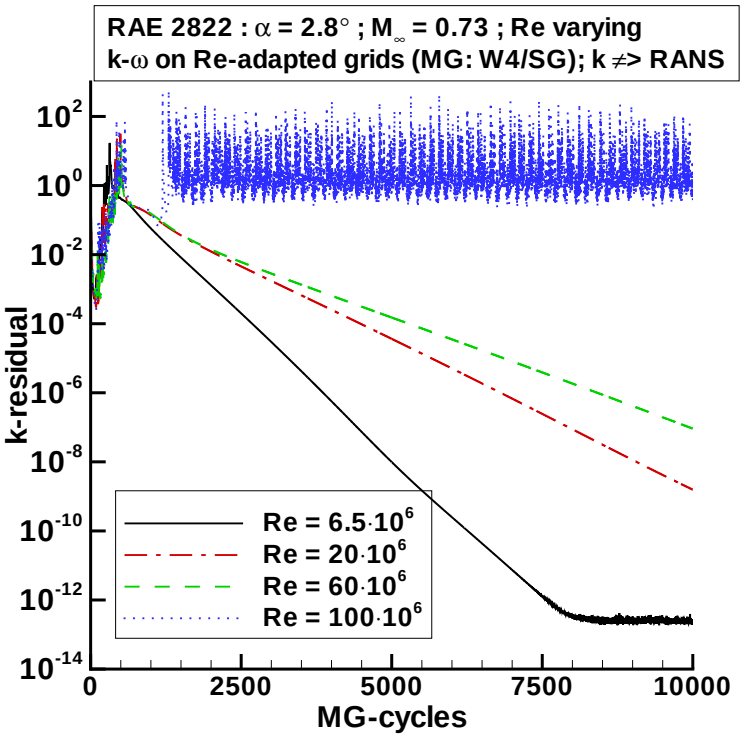

Figure 6.33: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with minimized coupling, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid ($\mathrm{Re} = 6.5 \cdot 10^6$: CFLTU $= 10$)

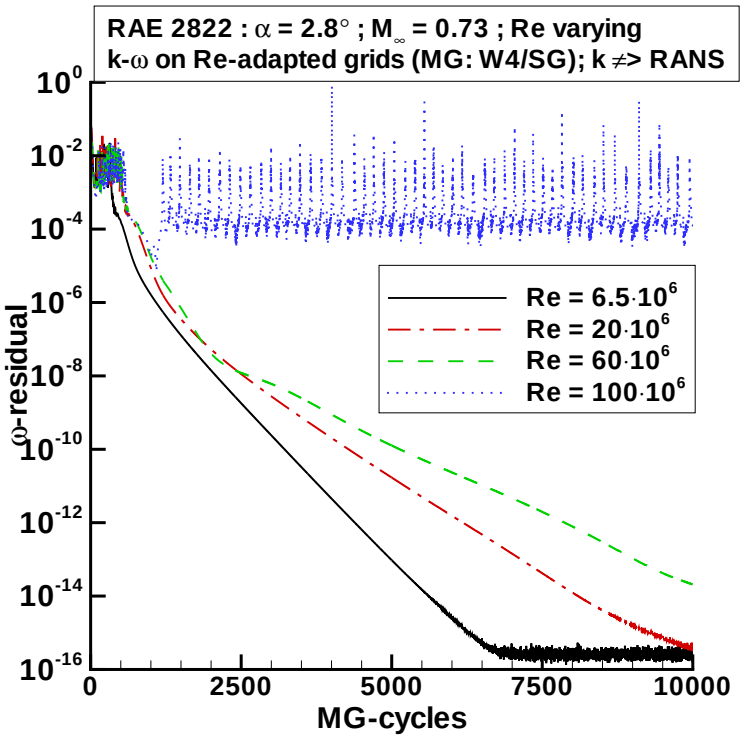

Figure 6.34: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with minimized coupling, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid ($\mathrm{Re} = 6.5 \cdot 10^6$: CFLTU $= 10$)

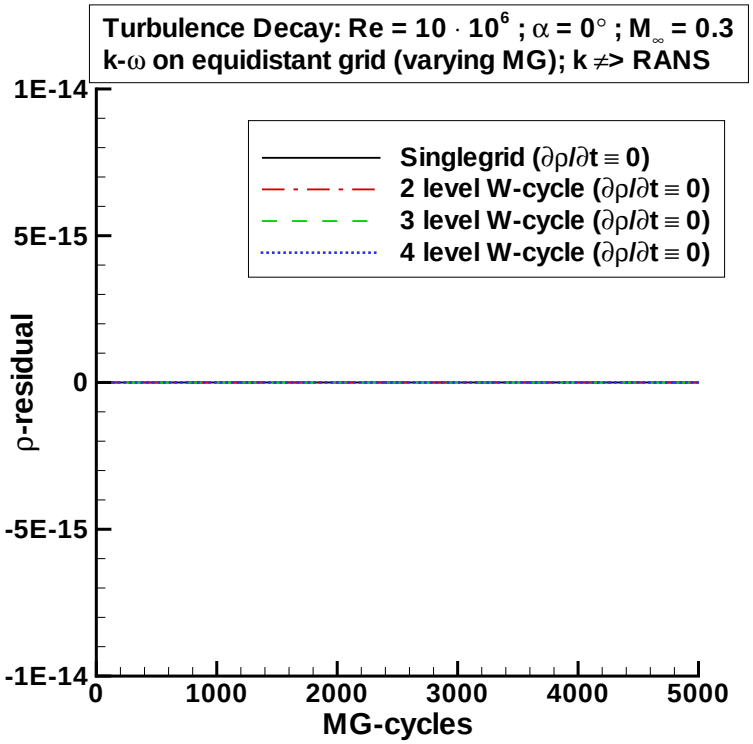

Figure 6.35: Convergence of density residual; turbulence decay, RANS calculation using k–ω turbulence model with minimized coupling, varying number of multigrid levels in use, W–cycle

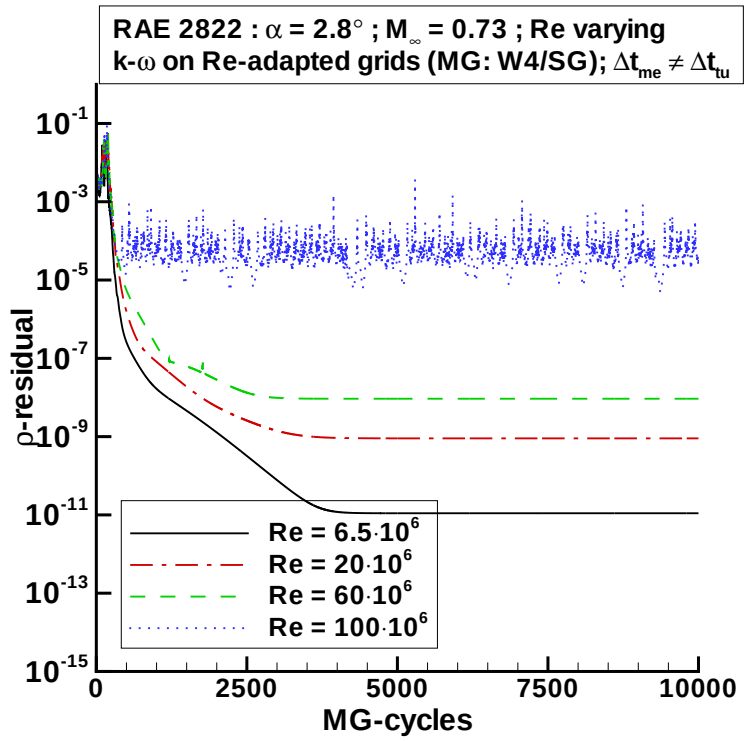

Figure 6.36: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

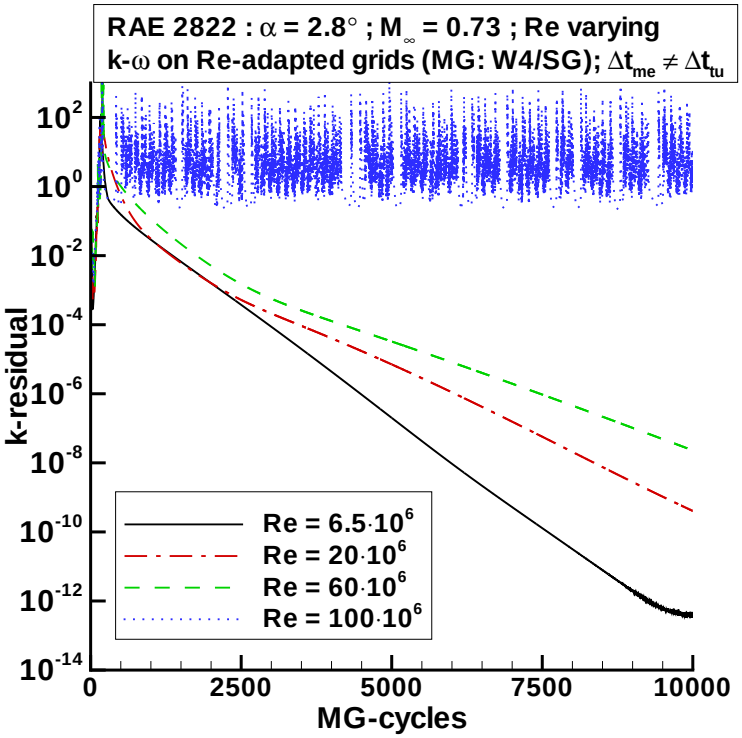

Figure 6.37: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

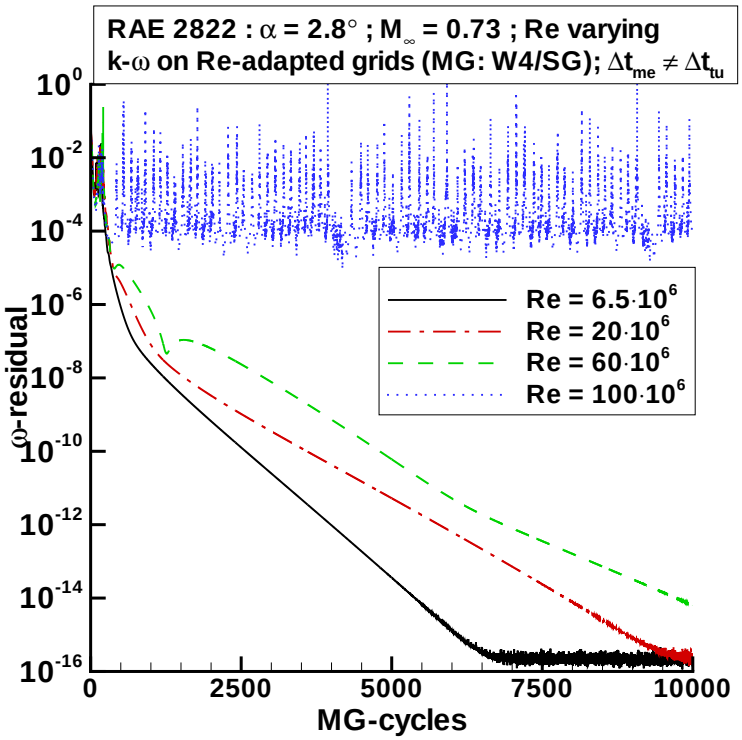

Figure 6.38: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

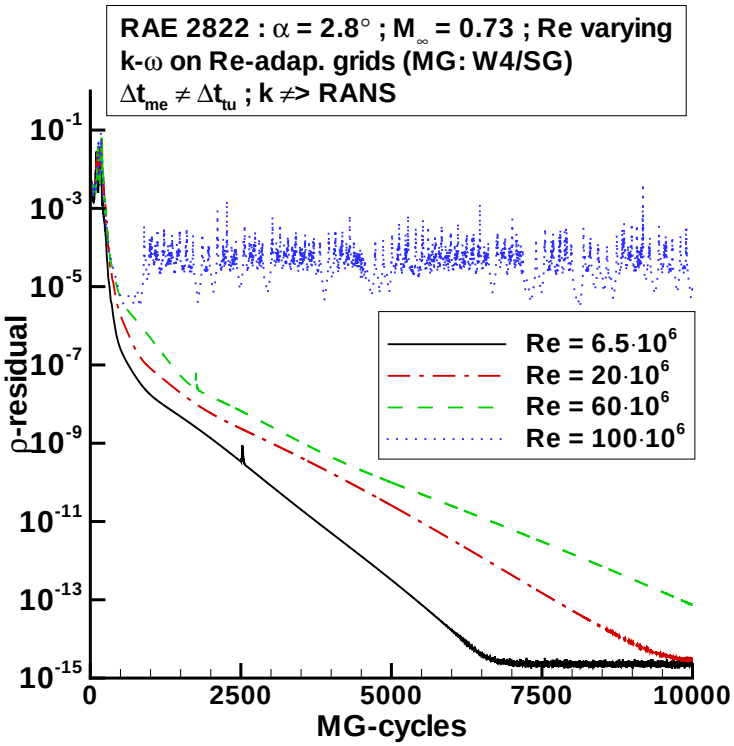

Figure 6.39: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations and minimized coupling, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

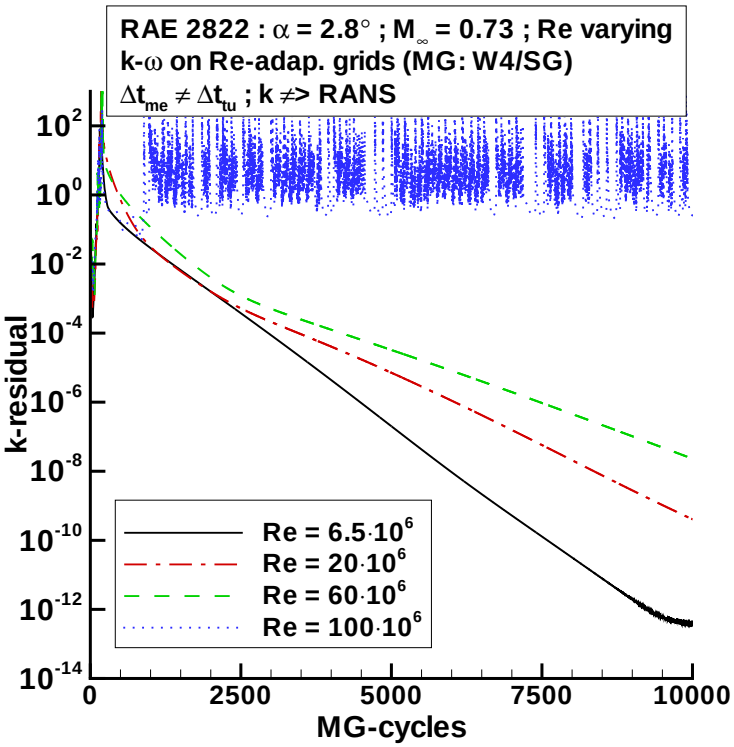

Figure 6.40: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations and minimized coupling, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

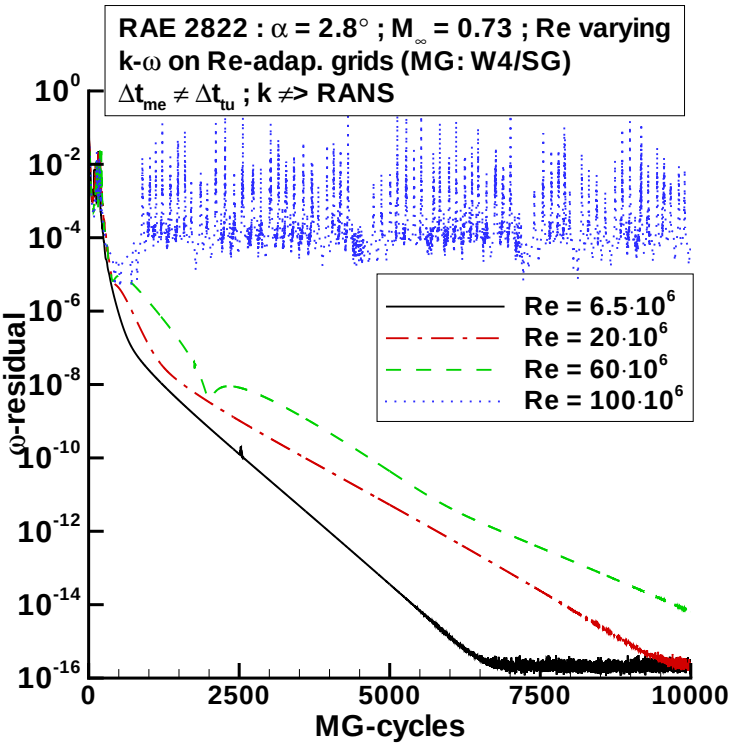

Figure 6.41: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations and minimized coupling, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

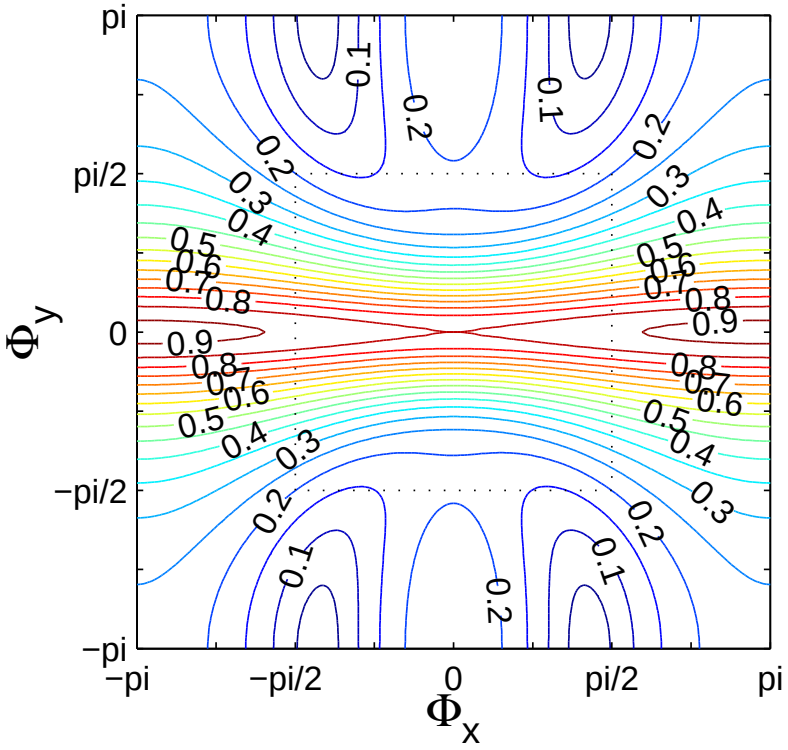

Figure 6.42: Fourier footprint of $g^{(q=5)}_{\text{alt. pi}-\text{src, IRS}}$ with $\alpha_{\texttt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = -1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$

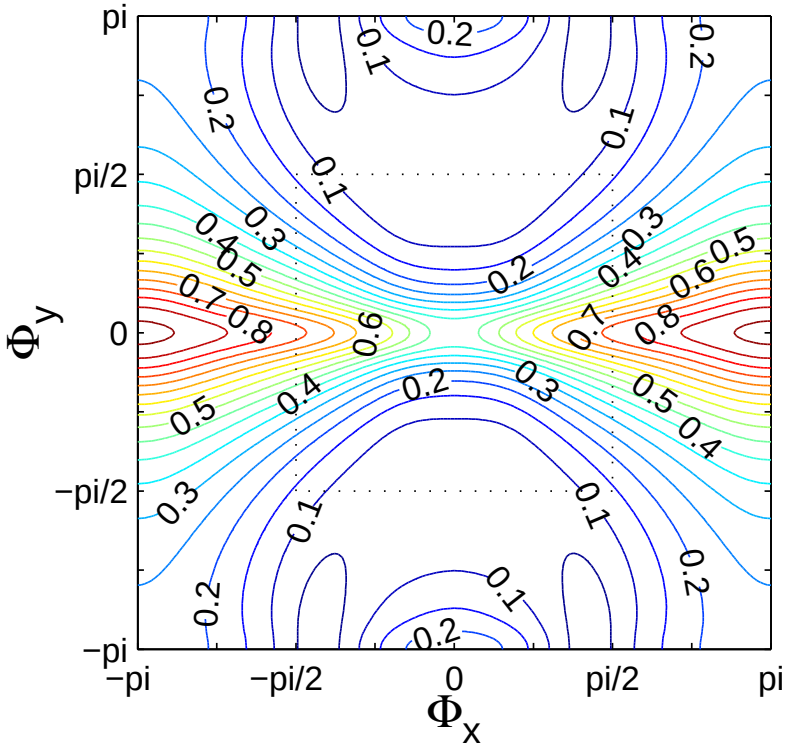

Figure 6.43: Fourier footprint of $g^{(q=5),\,2\mathrm{G}}_{\text{alt. pi}-\text{src, IRS}}$ with $\alpha_{\texttt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = -1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 2 multigrid level ($\mathcal{P}^H_{2H} = 1\ \forall H$)

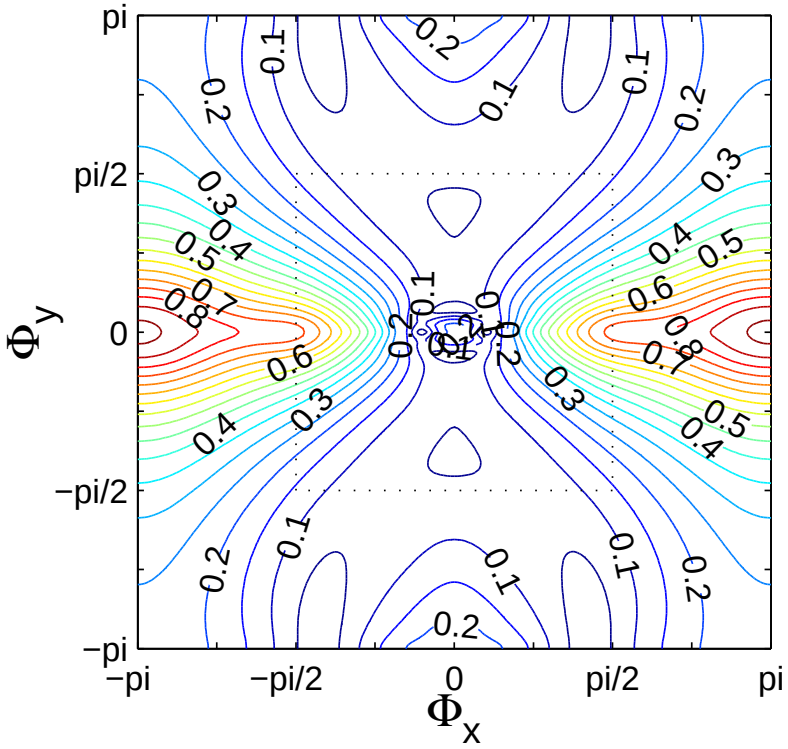

Figure 6.44: Fourier footprint of $g^{(q=5),\,\mathrm{MG}}_{\text{alt. pi}-\text{src, IRS}}$ with $\alpha_{\texttt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = -1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 3 multigrid level ($\mathcal{P}^H_{2H} = 1\ \forall H$)

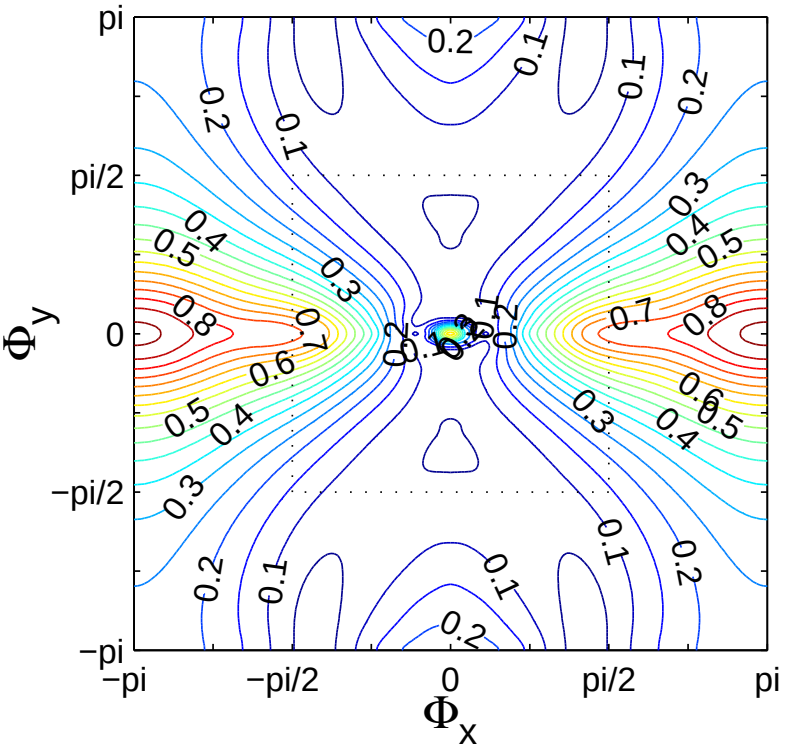

Figure 6.45: Fourier footprint of $g^{(q=5),\,\mathrm{MG}}_{\text{alt. pi}-\text{src, IRS}}$ with $\alpha_{\texttt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = -1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; 4 multigrid level ($\mathcal{P}^H_{2H} = 1\ \forall H$)

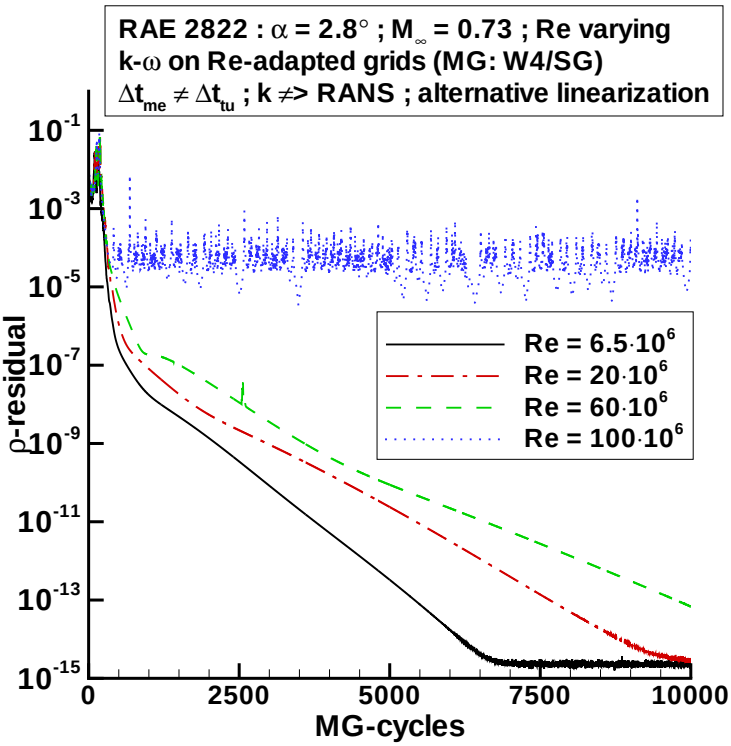

Figure 6.46: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations and minimized coupling, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid, alternative linearization of point implicit treatment of source term

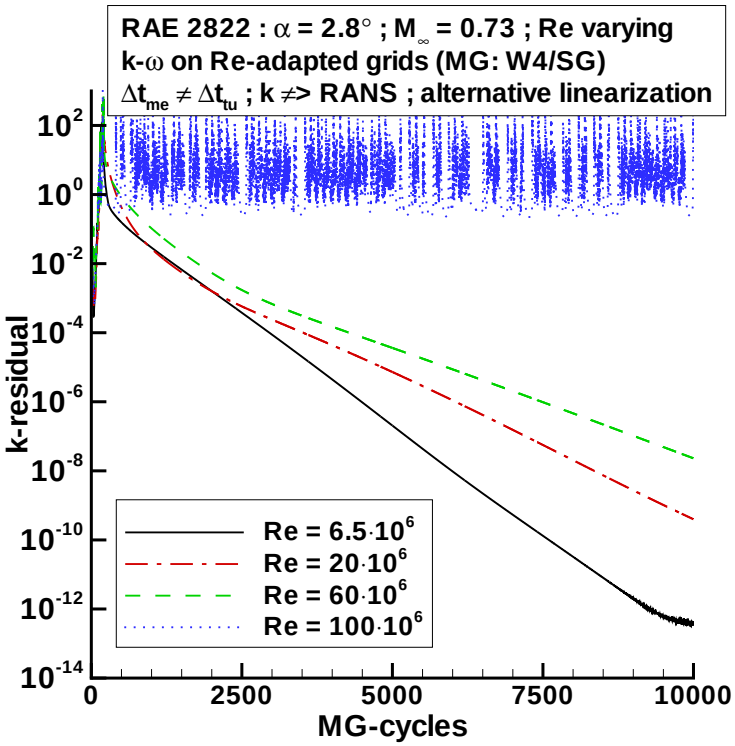

Figure 6.47: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations and minimized coupling, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid, alternative linearization of point implicit treatment of source term

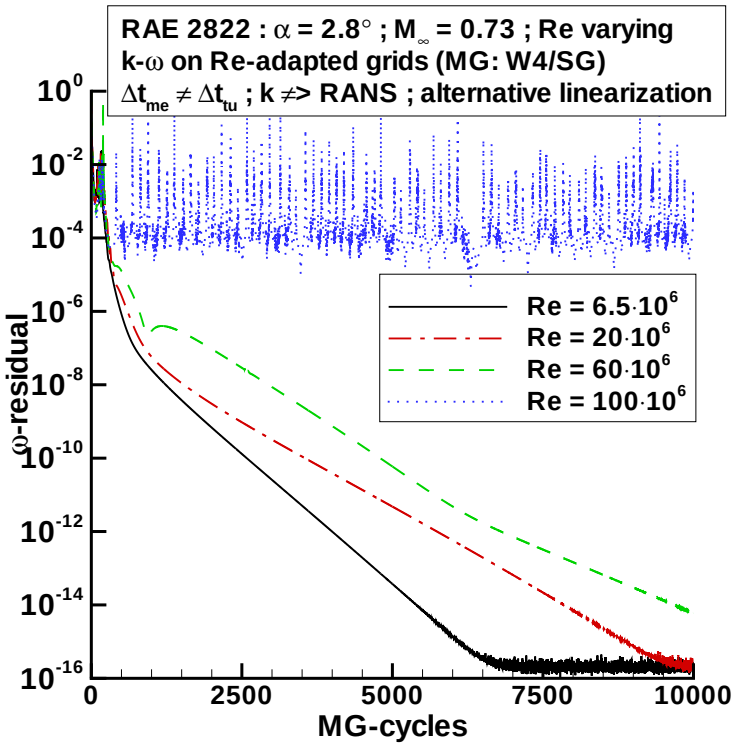

Figure 6.48: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations and minimized coupling, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid, alternative linearization of point implicit treatment of source term

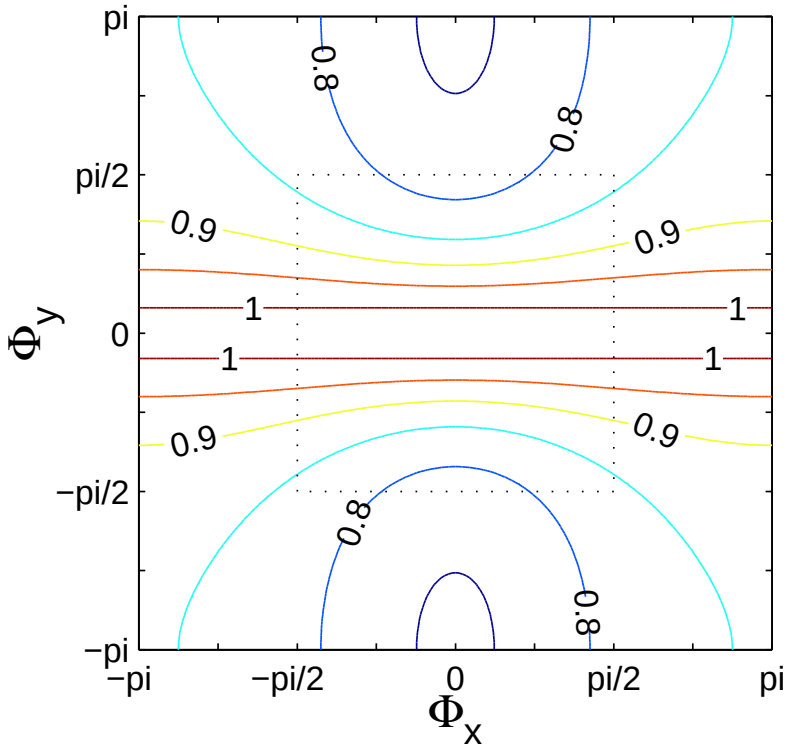

Figure 6.49: Fourier footprint of $g^{(q=5)}_{\mathrm{pi-eq,\,IRS}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; $\mathtt{CFL} = 7.5$

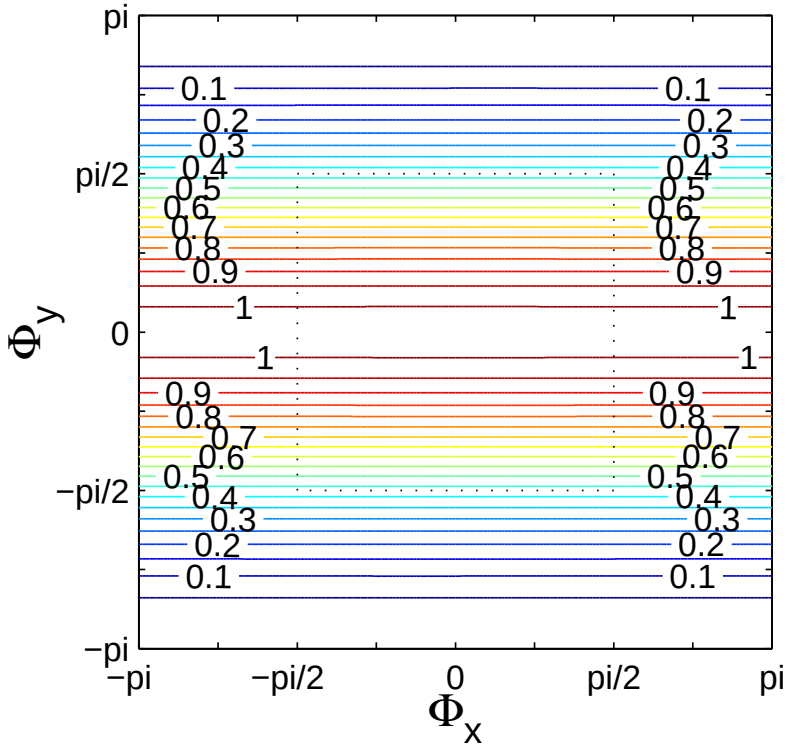

Figure 6.50: Fourier footprint of $g^{(q=5)}_{\mathrm{pi-eq}}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; no implicit residual smoothing, $\mathtt{CFL} = 3.75$

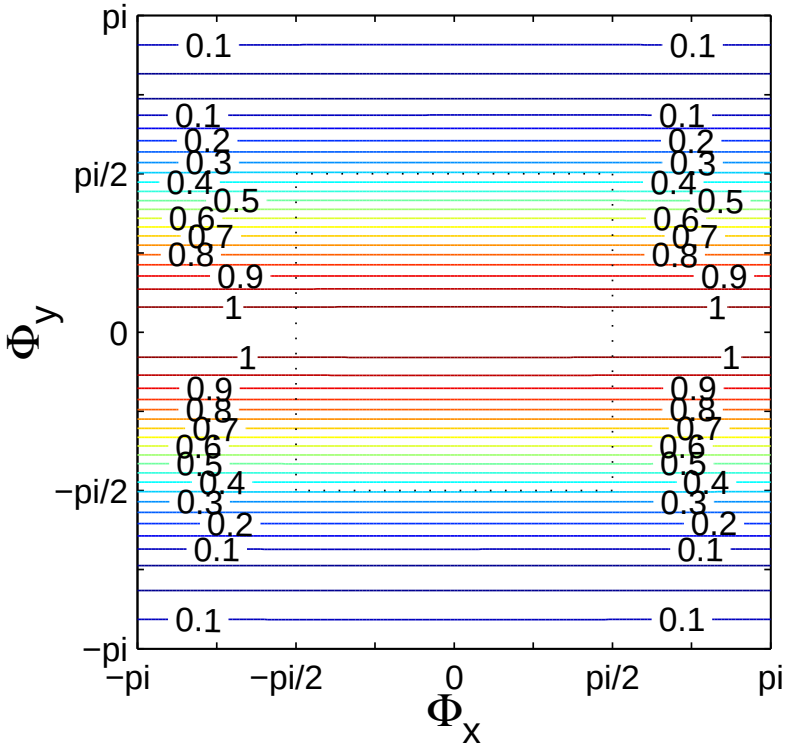

Figure 6.51: Fourier footprint of $g_{\mathrm{pi-eq}}^{(q=5)}$ with $\alpha_{\mathtt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; no implicit residual smoothing, $\mathtt{CFL} = 7.5$

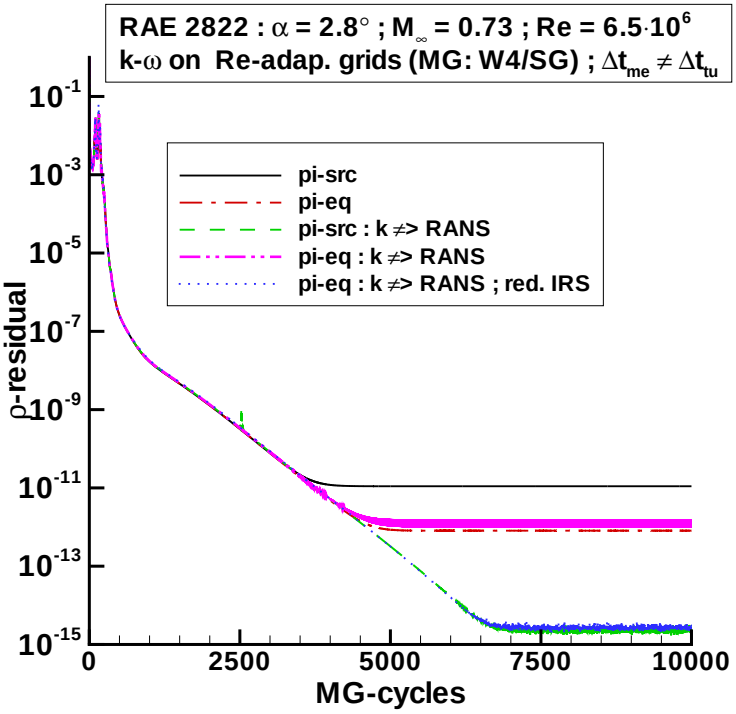

Figure 6.52: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, $\mathrm{Re} = 6.5 \cdot 10^6$, varying point implicit approaches; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

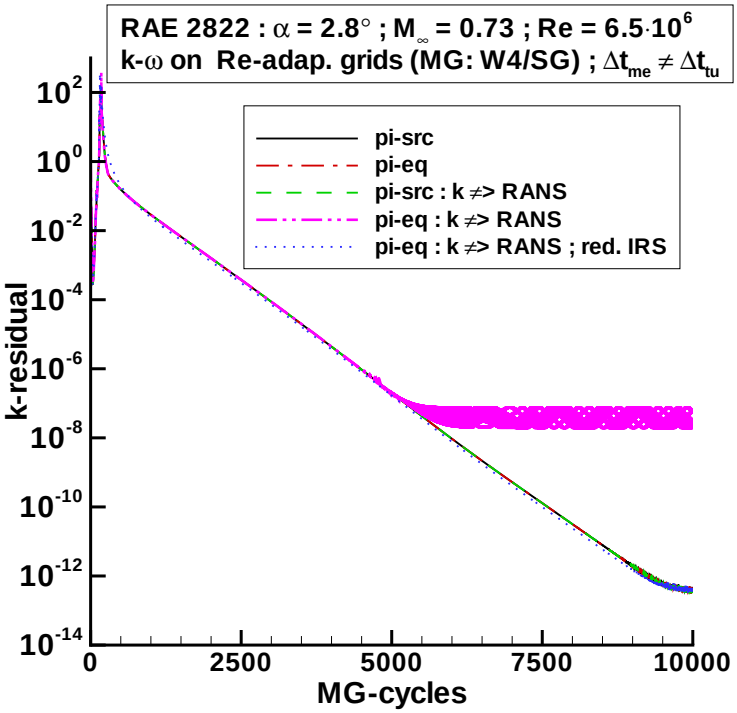

Figure 6.53: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, Re $= 6.5 \cdot 10^6$, varying point implicit approaches; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

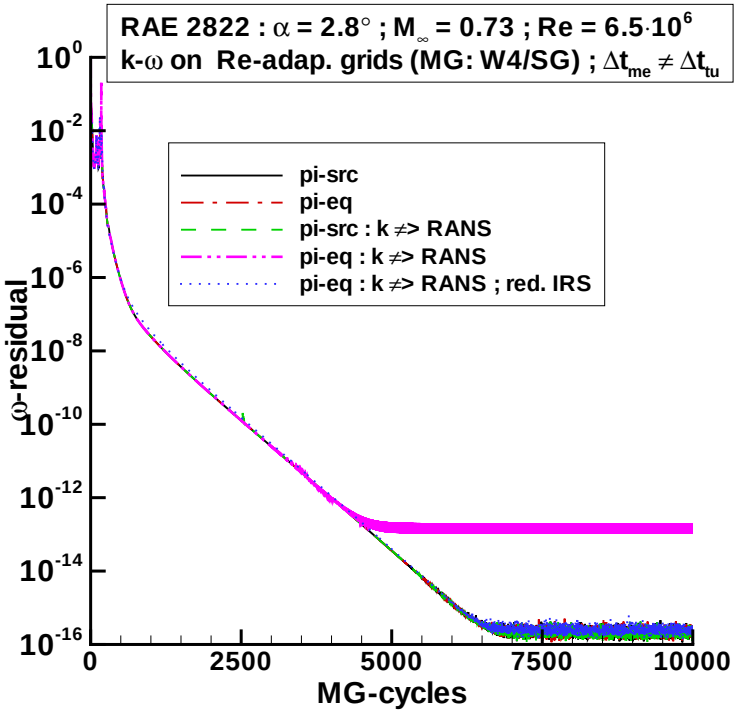

Figure 6.54: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, Re $= 6.5 \cdot 10^6$, varying point implicit approaches; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

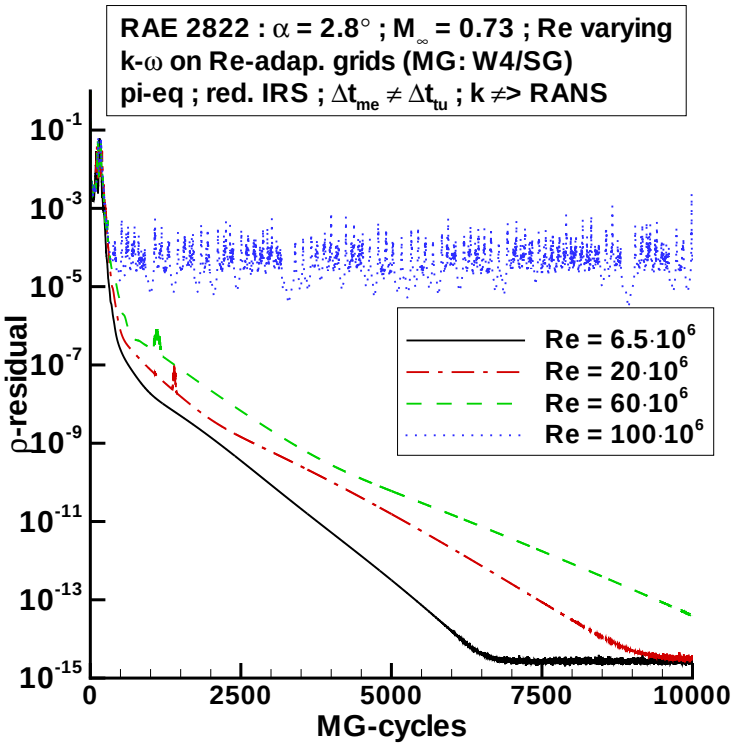

Figure 6.55: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, varying Reynolds number, point implicit treatment of complete turbulence equations, reduced implicit residual smoothing; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

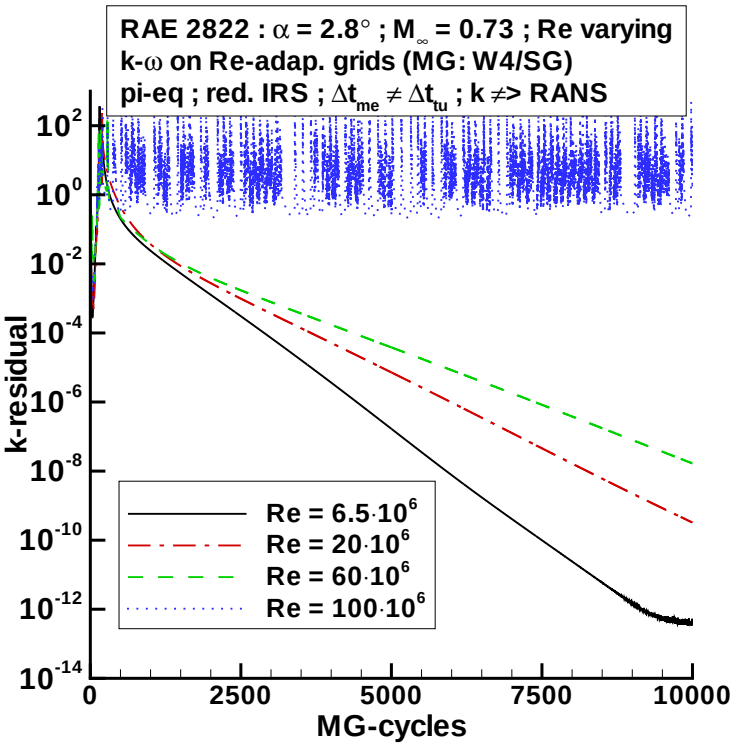

Figure 6.56: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, varying Reynolds number, point implicit treatment of complete turbulence equations, reduced implicit residual smoothing; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

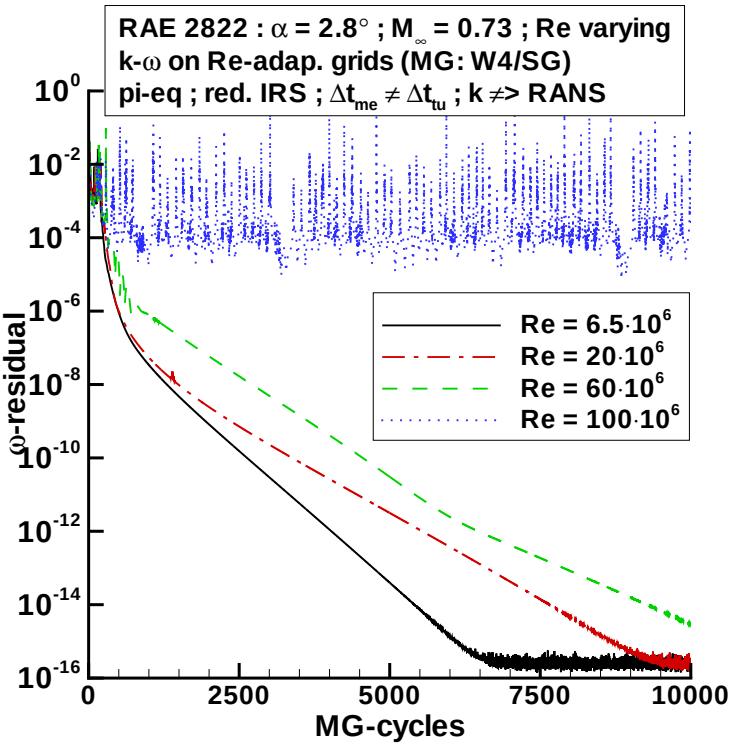

Figure 6.57: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, varying Reynolds number, point implicit treatment of complete turbulence equations, reduced implicit residual smoothing; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

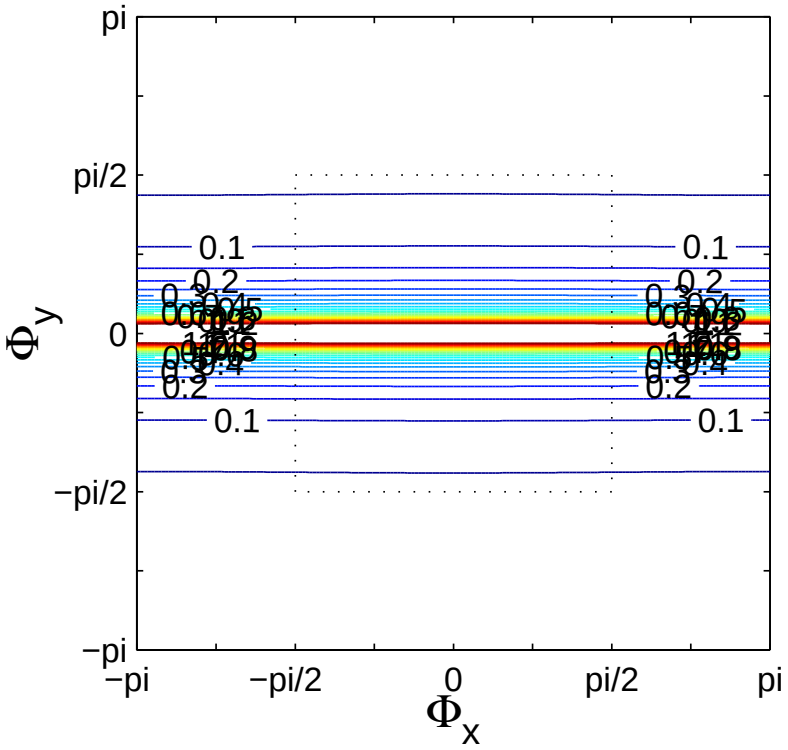

Figure 6.58: Fourier footprint of $g_{j\text{–line}}$ with $\alpha_{\texttt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; $\texttt{CFL} = 150$

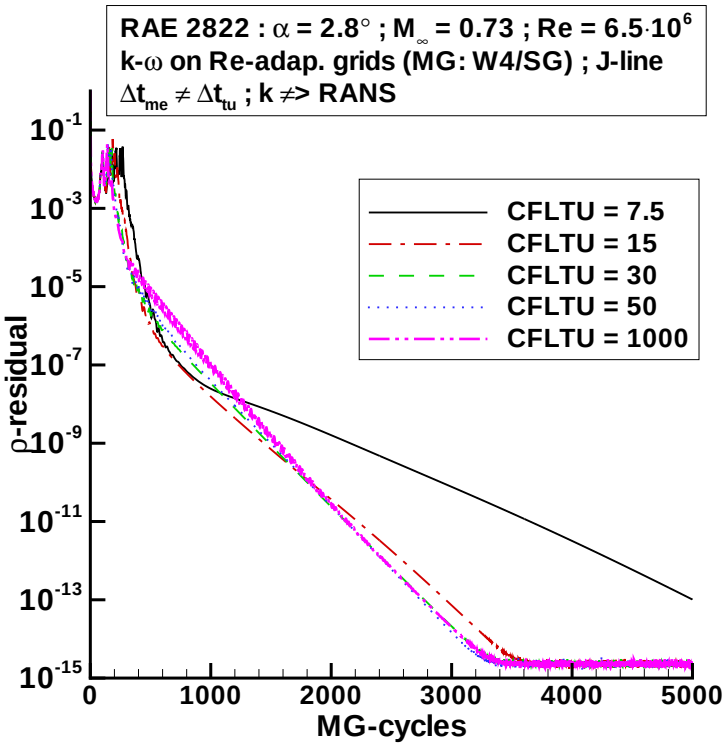

Figure 6.59: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, Re $= 6.5 \cdot 10^6$, j–line implicit, varying CFLTU; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

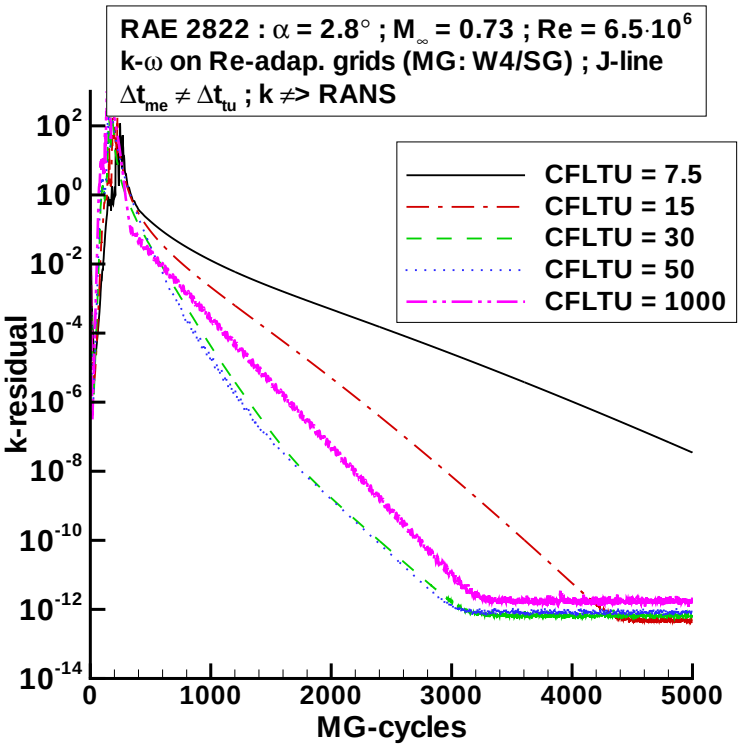

Figure 6.60: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, Re $= 6.5 \cdot 10^6$, j–line implicit, varying CFLTU; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

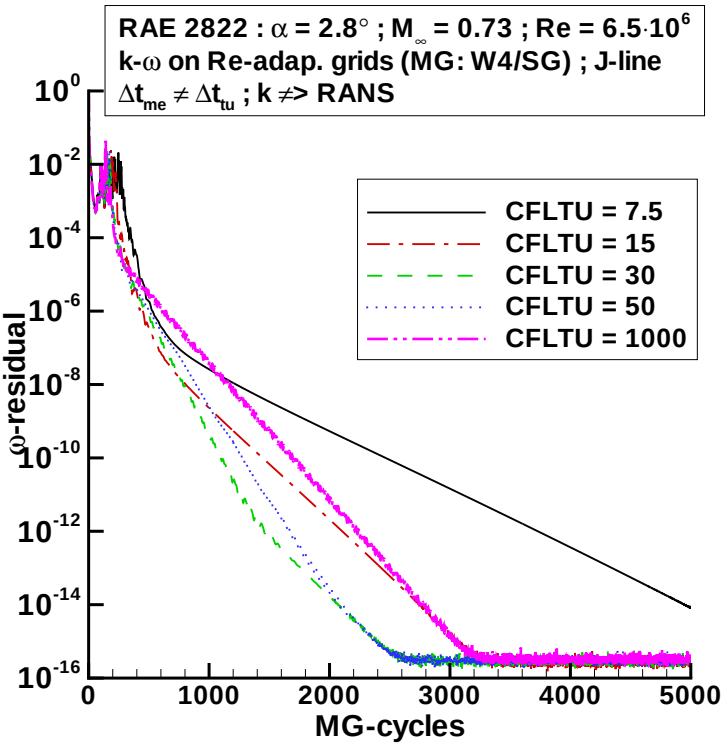

Figure 6.61: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, Re $= 6.5 \cdot 10^6$, j–line implicit, varying CFLTU; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

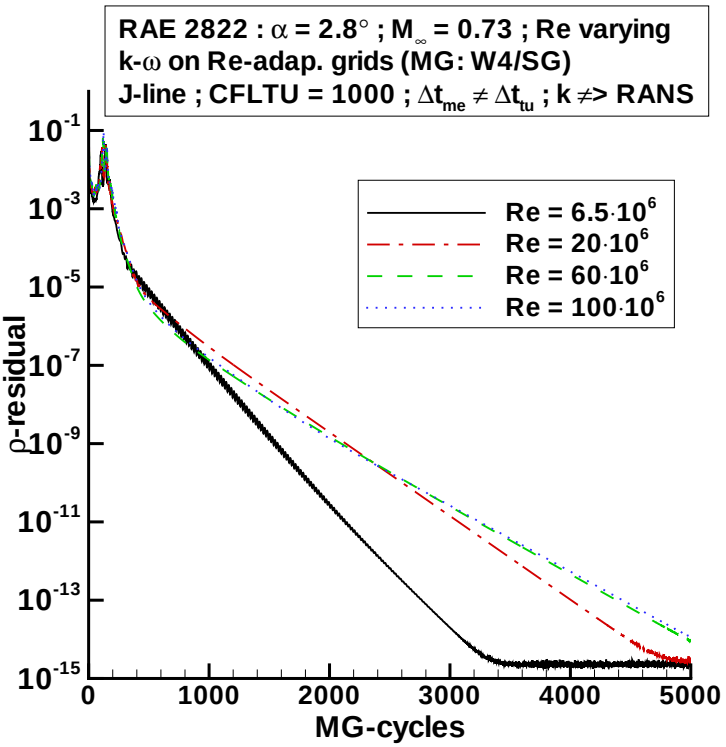

Figure 6.62: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, j–line implicit, CFLTU $=$ 1000, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

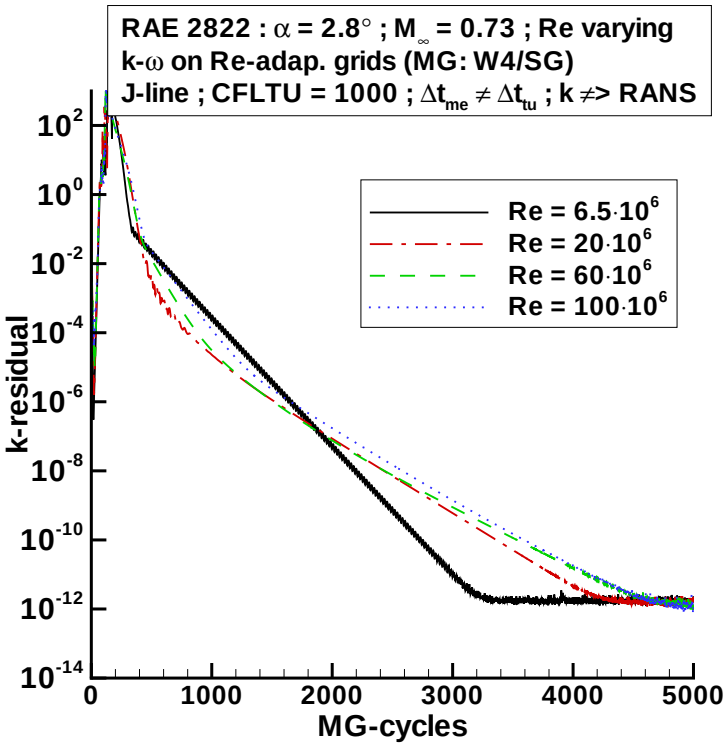

Figure 6.63: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, j–line implicit, CFLTU $= 1000$, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

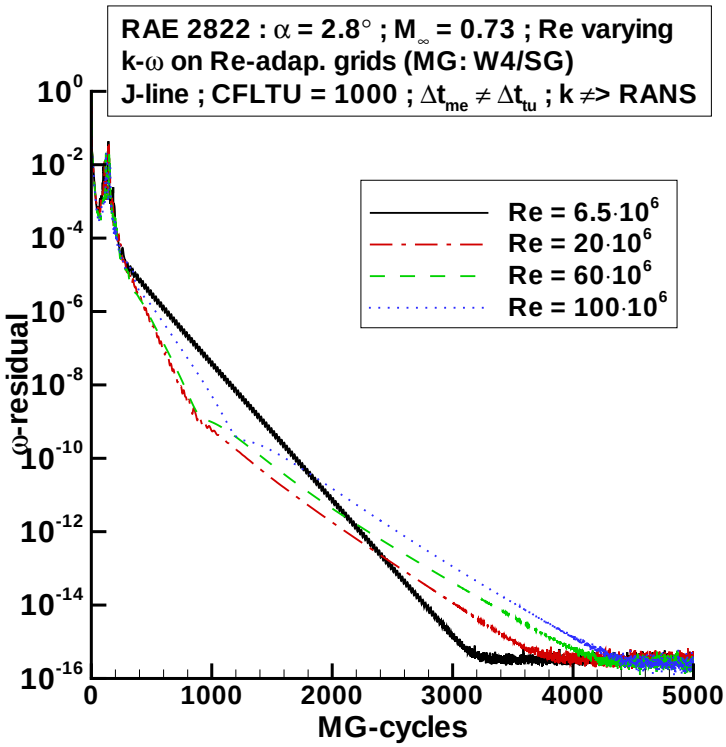

Figure 6.64: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, j–line implicit, CFLTU $= 1000$, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

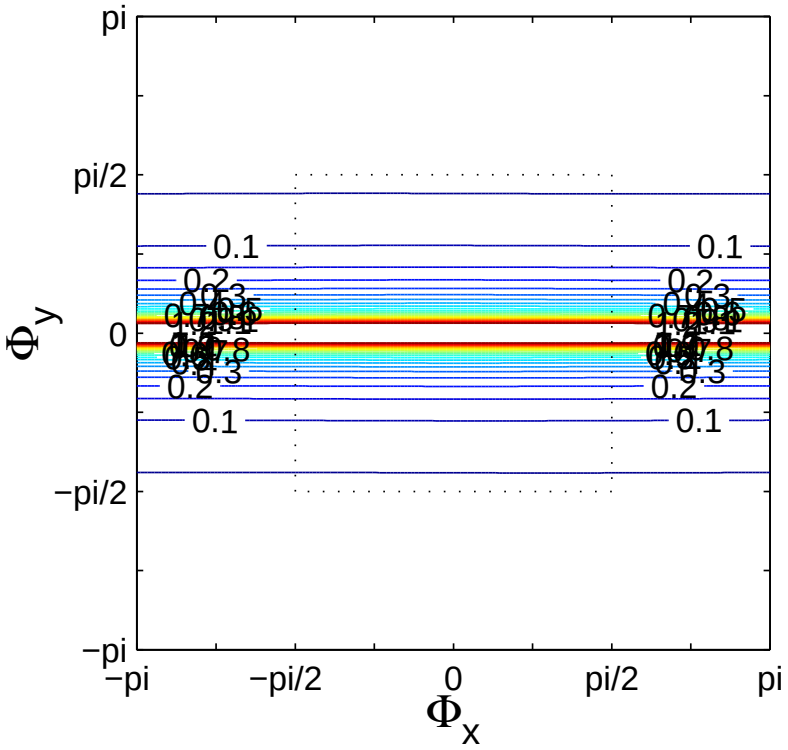

Figure 6.65: Fourier footprint of g_{DDADI} with $\alpha_{\texttt{cent}}$ applied to Eq. (4.1) [$u = 10^{-1}$; $v = 10^{-2}$; $\Delta x = 10^{-2}$; $\Delta y = 10^{-6}$; $\mu = 10^{-8}$; $S = 1000$]: convection diffusion equation including source term with $\Delta x/\Delta y = 10000$ and $(v\Delta x)/(u\Delta y) = 1000$; $\texttt{CFL} = 150$

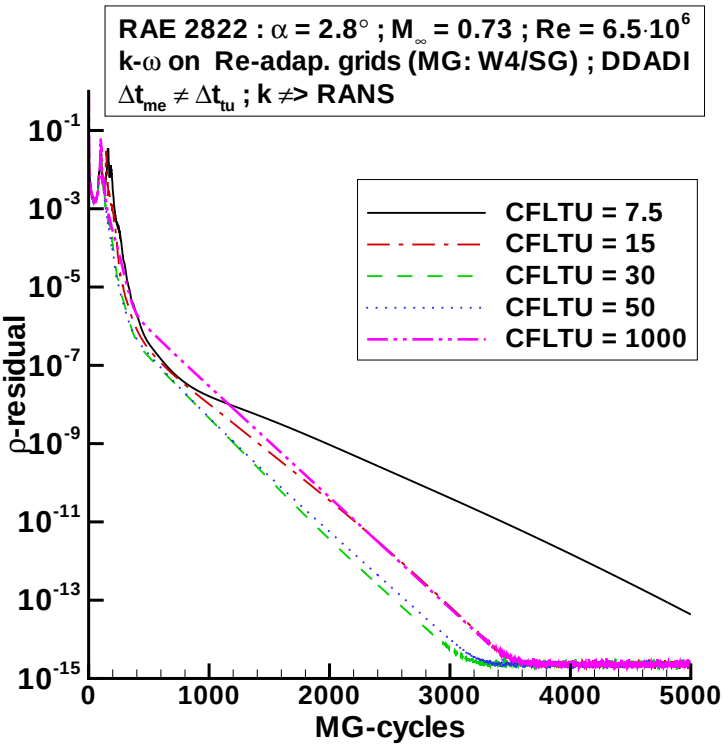

Figure 6.66: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, $\mathrm{Re} = 6.5 \cdot 10^6$, DDADI, varying $\texttt{CFLTU}$; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

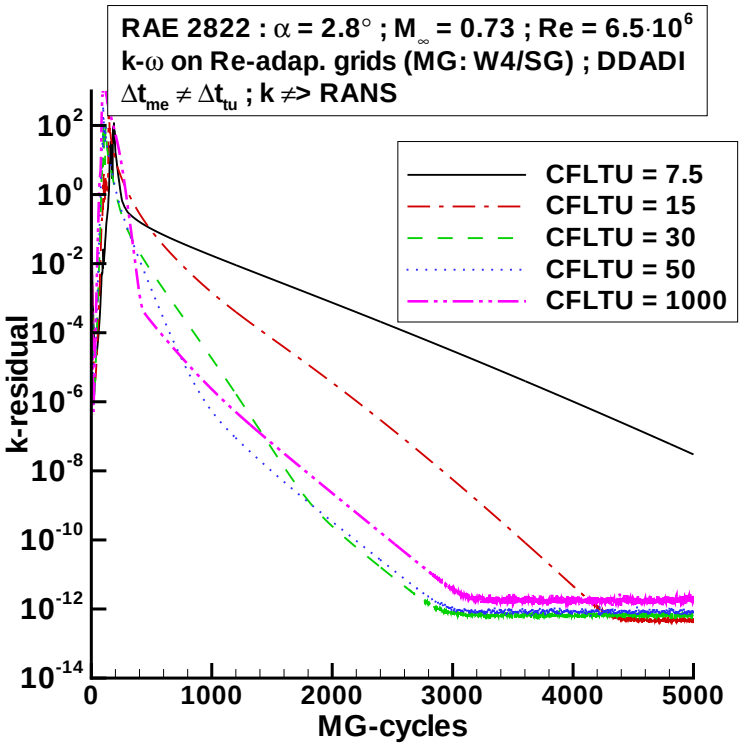

Figure 6.67: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, $\mathrm{Re} = 6.5 \cdot 10^6$, DDADI, varying CFLTU; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

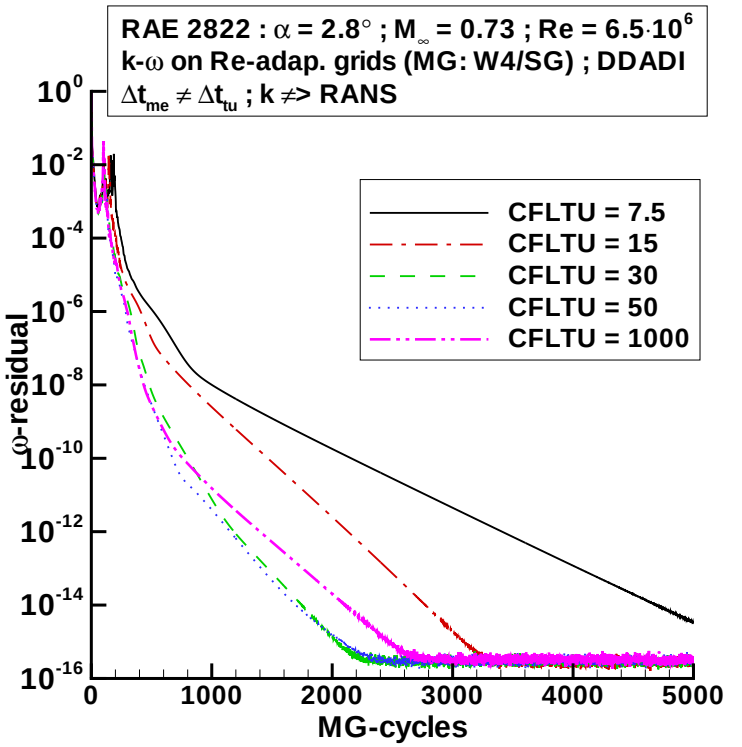

Figure 6.68: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, $\mathrm{Re} = 6.5 \cdot 10^6$, DDADI, varying CFLTU; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

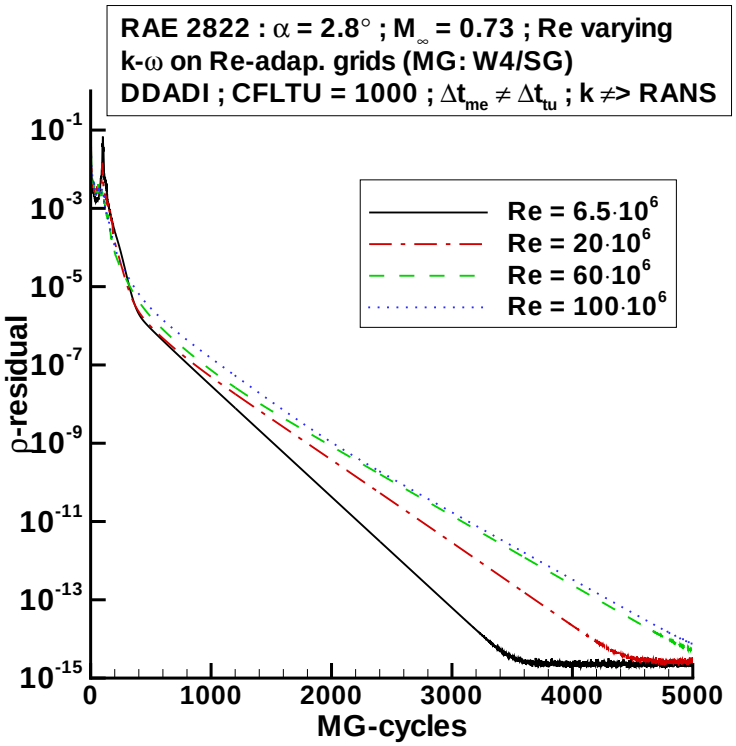

Figure 6.69: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, DDADI, CFLTU $= 1000$, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

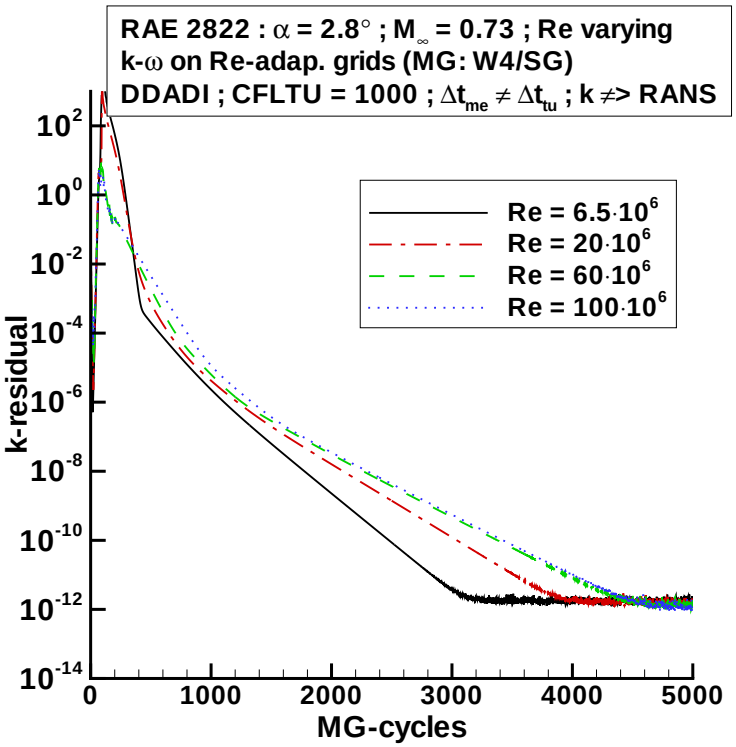

Figure 6.70: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, DDADI, CFLTU $= 1000$, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

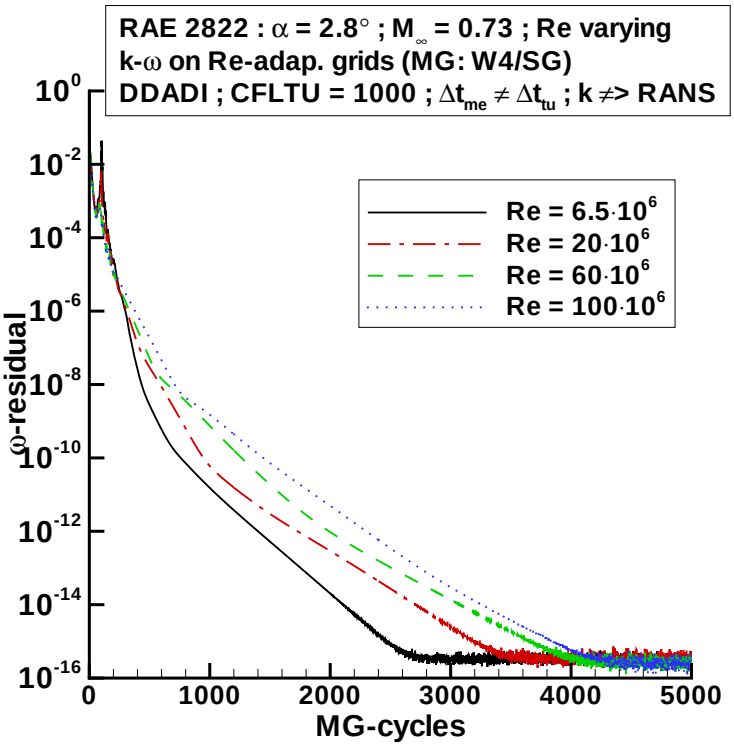

Figure 6.71: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, DDADI, $\texttt{CFLTU} = 1000$, varying Reynolds number; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

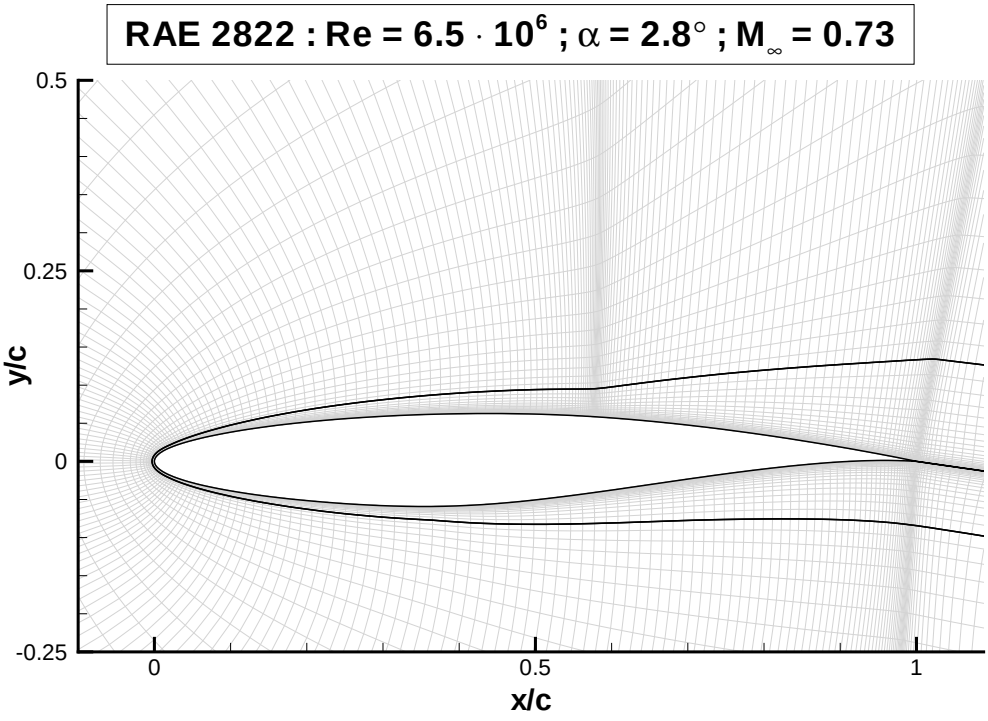

Figure 6.72: Computational grid for RAE 2822 airfoil at $\mathrm{Re} = 6.5 \cdot 10^6$ with 2 blocks

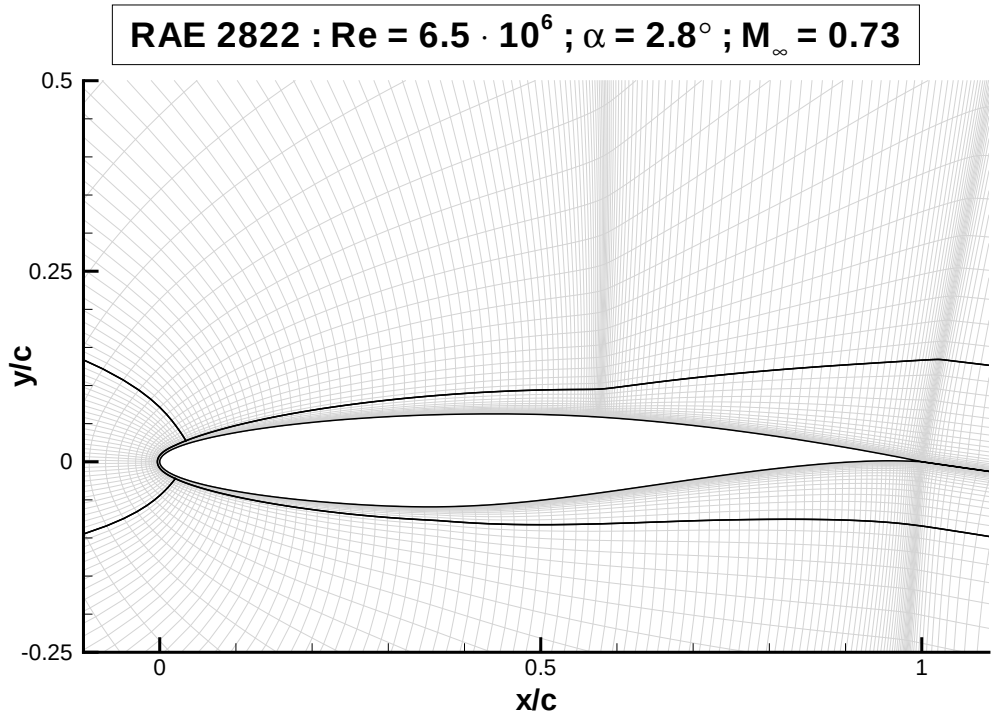

Figure 6.73: Computational grid for RAE 2822 airfoil at $\mathrm{Re} = 6.5 \cdot 10^6$ with 4 blocks

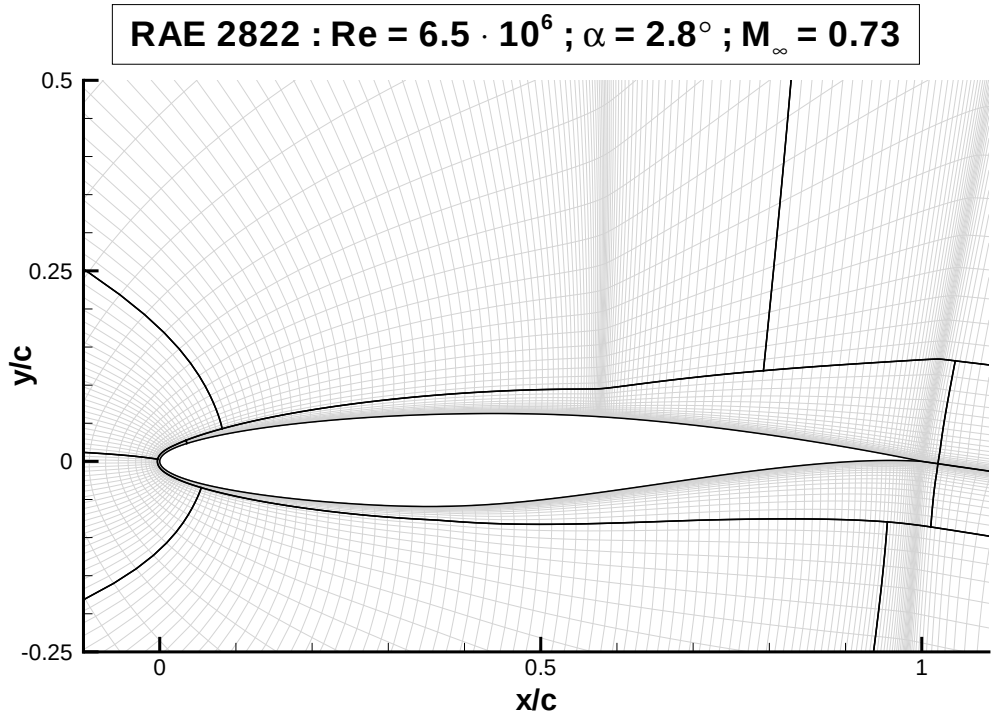

Figure 6.74: Computational grid for RAE 2822 airfoil at $\mathrm{Re} = 6.5 \cdot 10^6$ with 16 blocks

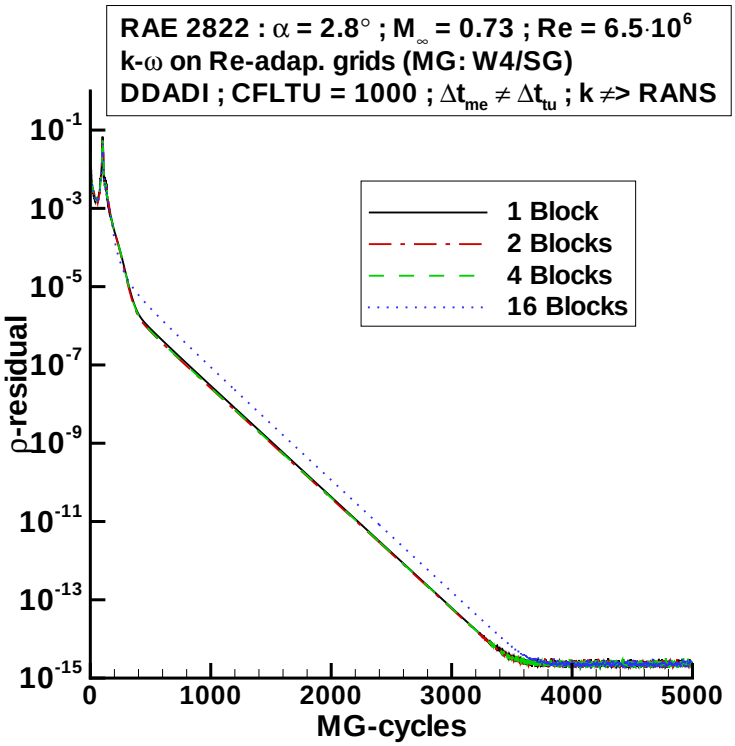

Figure 6.75: Convergence of density residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, $\mathrm{Re} = 6.5 \cdot 10^6$, DDADI, $\mathtt{CFLTU} = 1000$, varying number of blocks; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

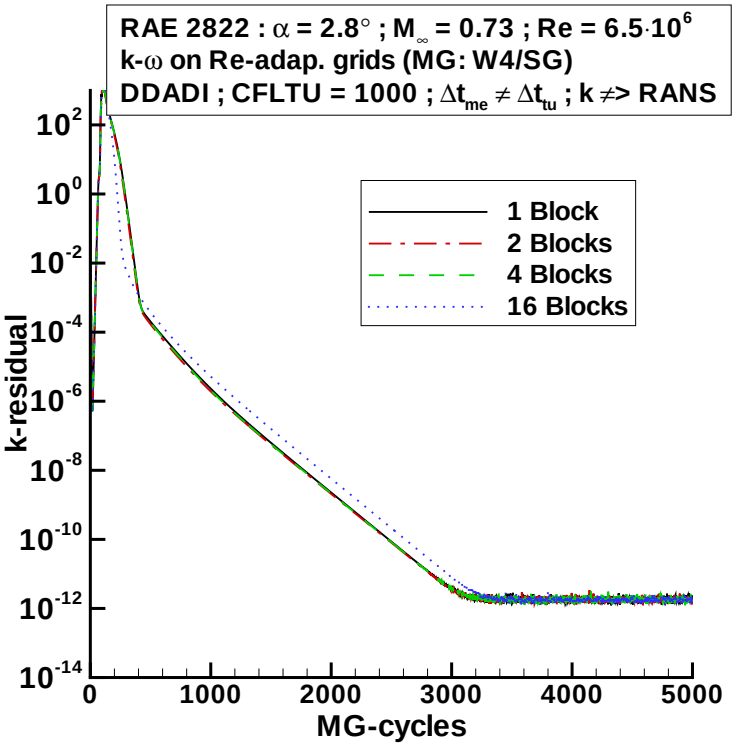

Figure 6.76: Convergence of k residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, $\mathrm{Re} = 6.5 \cdot 10^6$, DDADI, $\mathtt{CFLTU} = 1000$, varying number of blocks; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

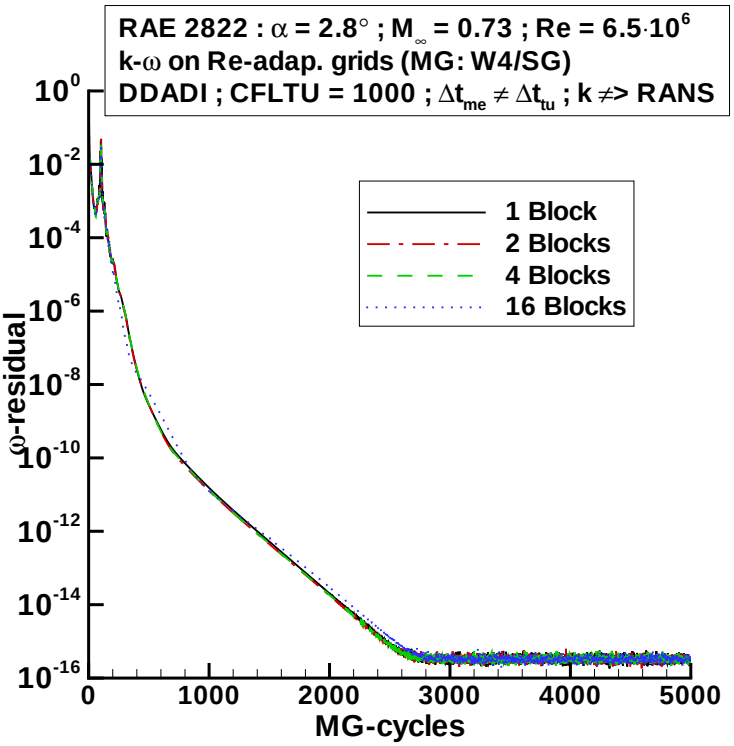

Figure 6.77: Convergence of ω residual; RAE 2822 airfoil, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, $\mathrm{Re} = 6.5 \cdot 10^6$, DDADI, $\mathtt{CFLTU} = 1000$, varying number of blocks; RANS: 4 multigrid level, W-cycle; k–ω: singlegrid

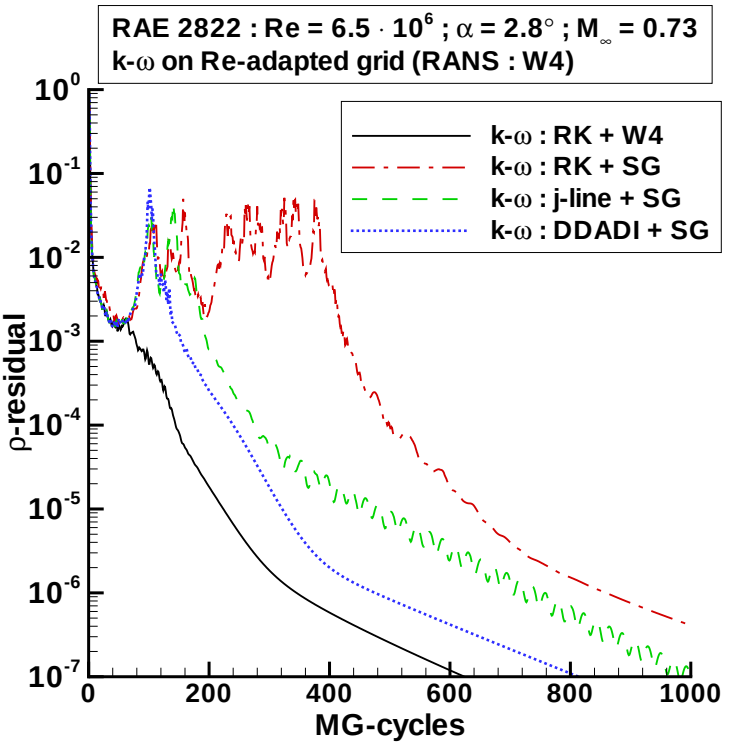

Figure 6.78: Convergence of density residual; RAE 2822 airfoil, $\mathrm{Re} = 6.5 \cdot 10^6$, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations ($\mathtt{CFLTU}_{\mathrm{expl}} = 7.5$; $\mathtt{CFLTU}_{\mathrm{impl}} = 1000$)

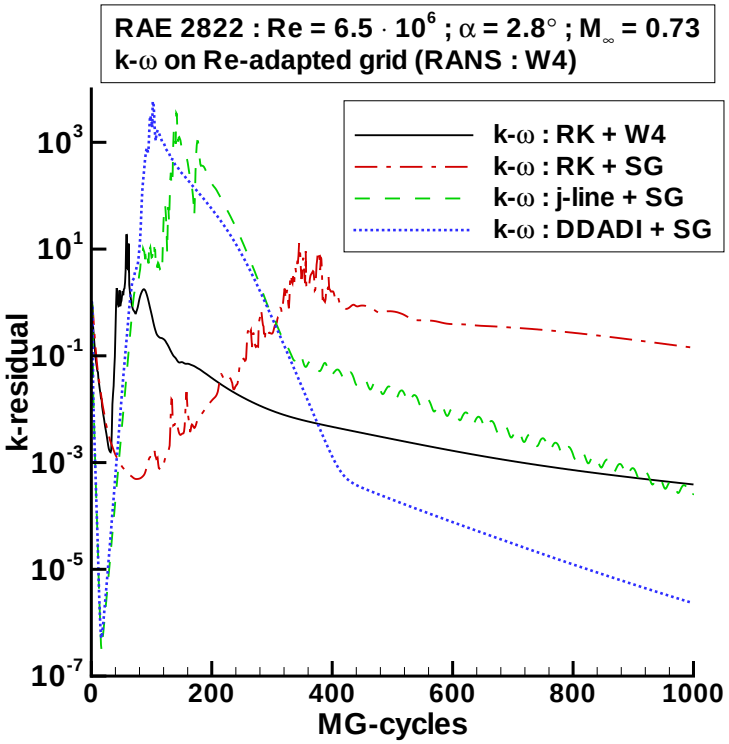

Figure 6.79: Convergence of k residual; RAE 2822 airfoil, Re $= 6.5 \cdot 10^6$, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations ($\texttt{CFLTU}_{\text{expl}} = 7.5$; $\texttt{CFLTU}_{\text{impl}} = 1000$)

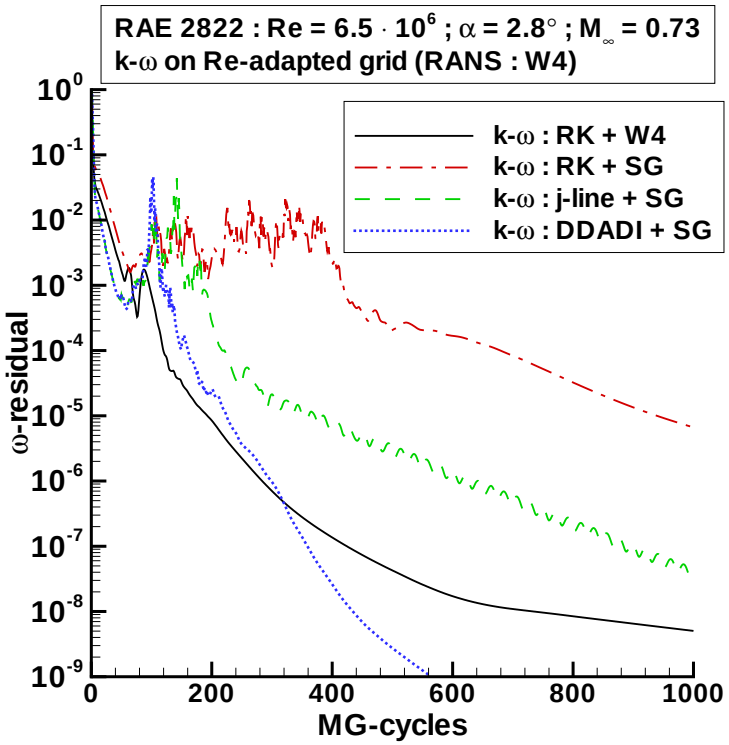

Figure 6.80: Convergence of ω residual; RAE 2822 airfoil, Re $= 6.5 \cdot 10^6$, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations ($\texttt{CFLTU}_{\text{expl}} = 7.5$; $\texttt{CFLTU}_{\text{impl}} = 1000$)

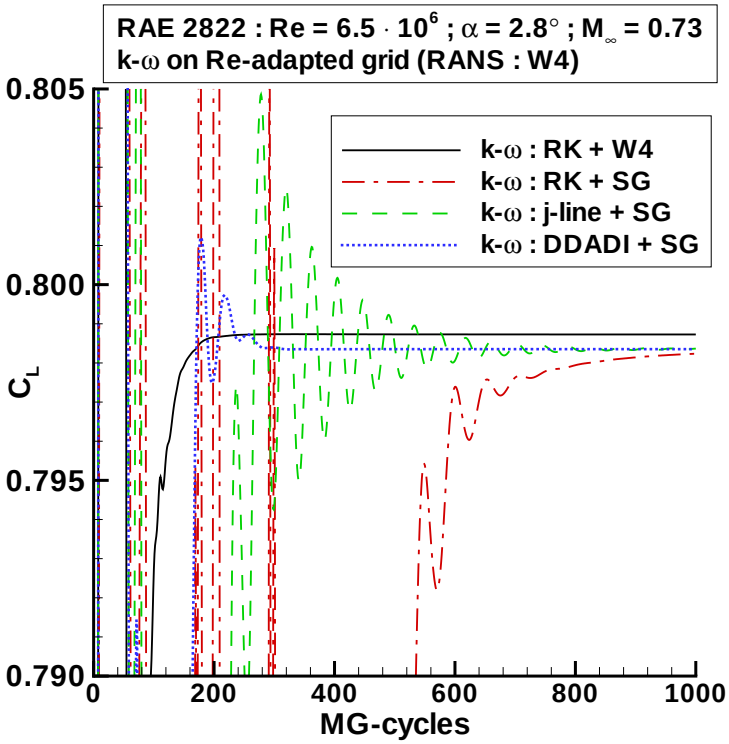

Figure 6.81: Convergence of lift coefficient; RAE 2822 airfoil, Re $= 6.5 \cdot 10^6$, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations ($\texttt{CFLTU}_{\text{expl}} = 7.5$; $\texttt{CFLTU}_{\text{impl}} = 1000$)

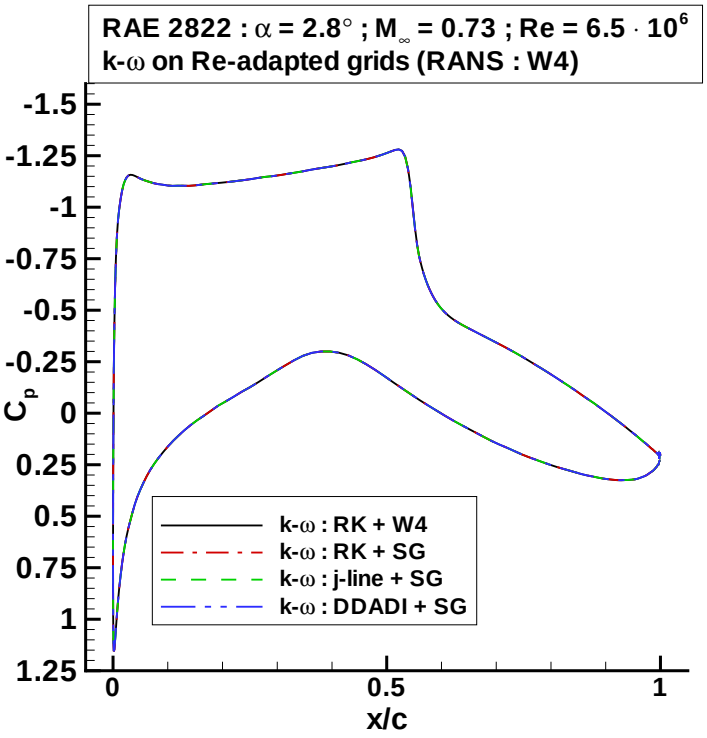

Figure 6.82: C_p distribution; RAE 2822 airfoil, Re $= 6.5 \cdot 10^6$, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations ($\texttt{CFLTU}_{\text{expl}} = 7.5$; $\texttt{CFLTU}_{\text{impl}} = 1000$)

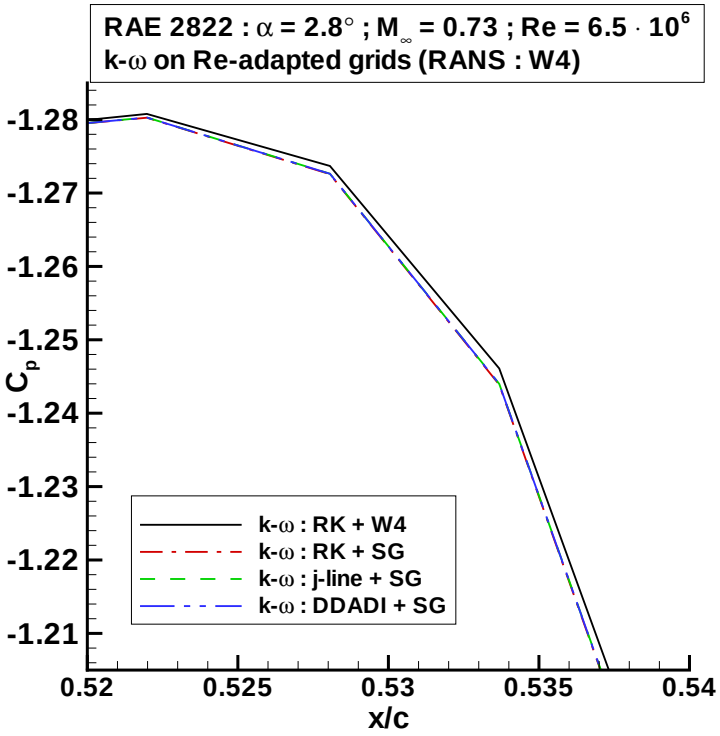

Figure 6.83: Close–up of C_p distribution; RAE 2822 airfoil, Re $= 6.5 \cdot 10^6$, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations ($\texttt{CFLTU}_{\text{expl}} = 7.5$; $\texttt{CFLTU}_{\text{impl}} = 1000$)

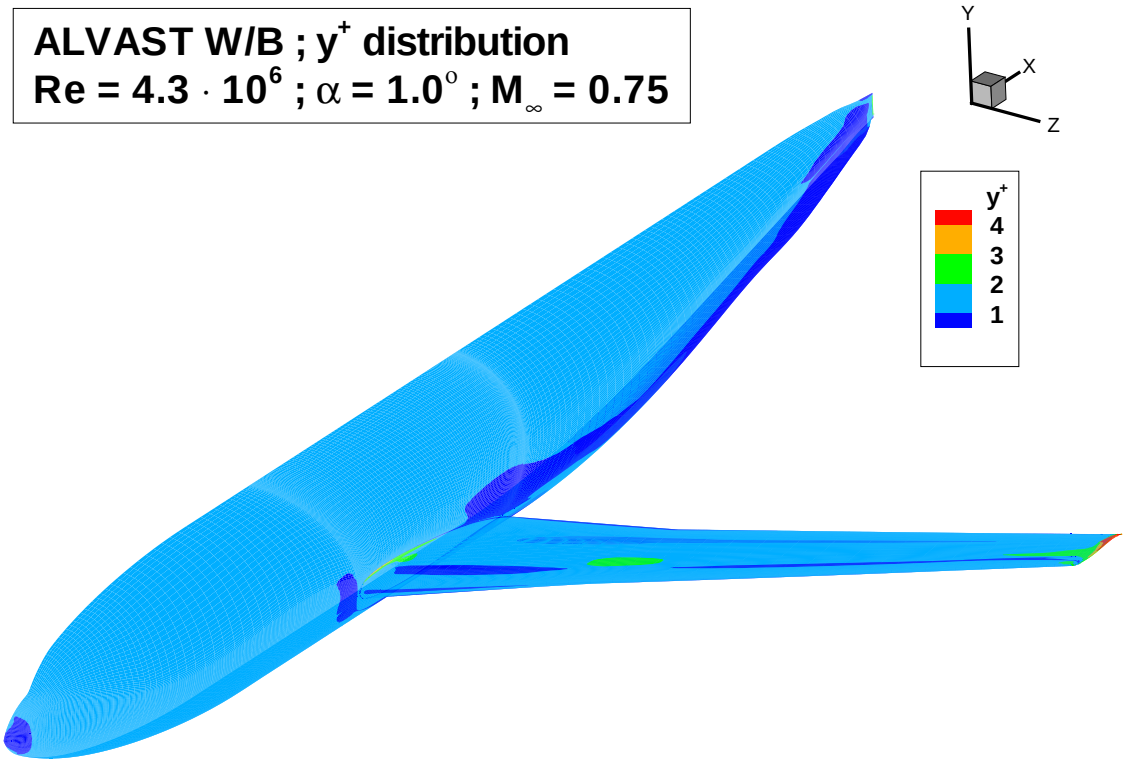

Figure 7.1: y_w^+ distribution; ALVAST Wing/Body configuration, RANS calculation using k–ω turbulence model, Re $= 4.3 \cdot 10^6$

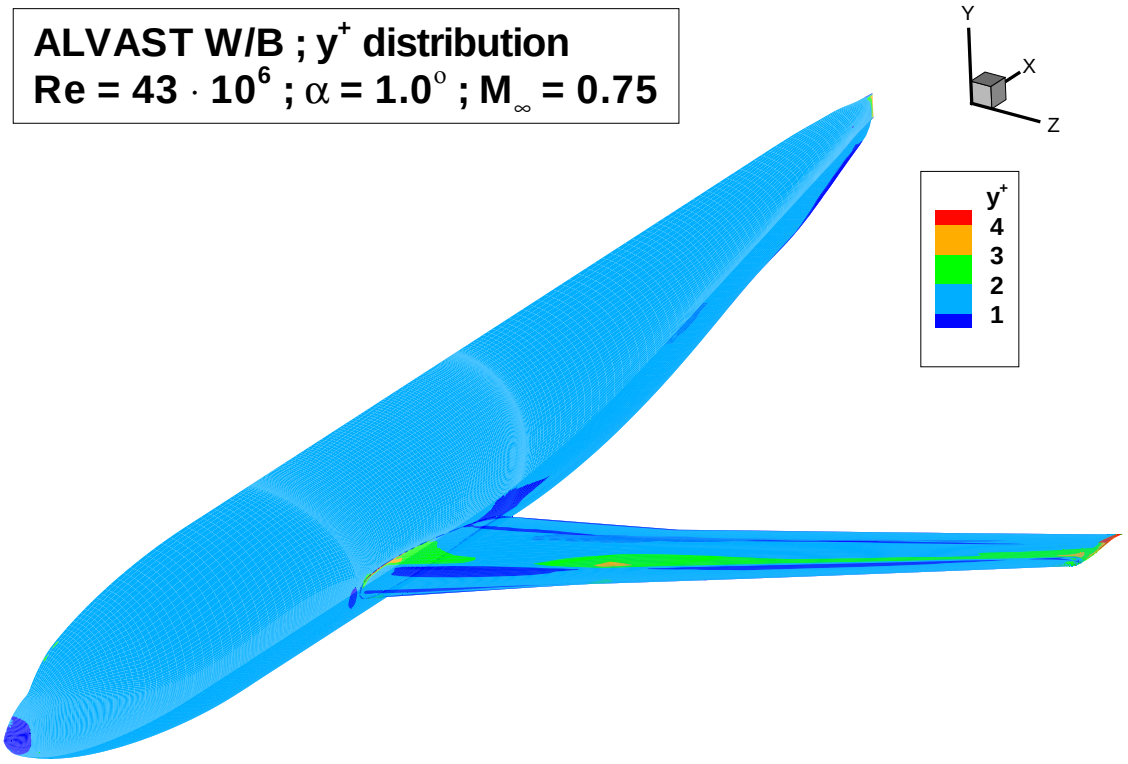

Figure 7.2: y_w^+ distribution; ALVAST Wing/Body configuration, RANS calculation using k–ω turbulence model, Re $= 43 \cdot 10^6$

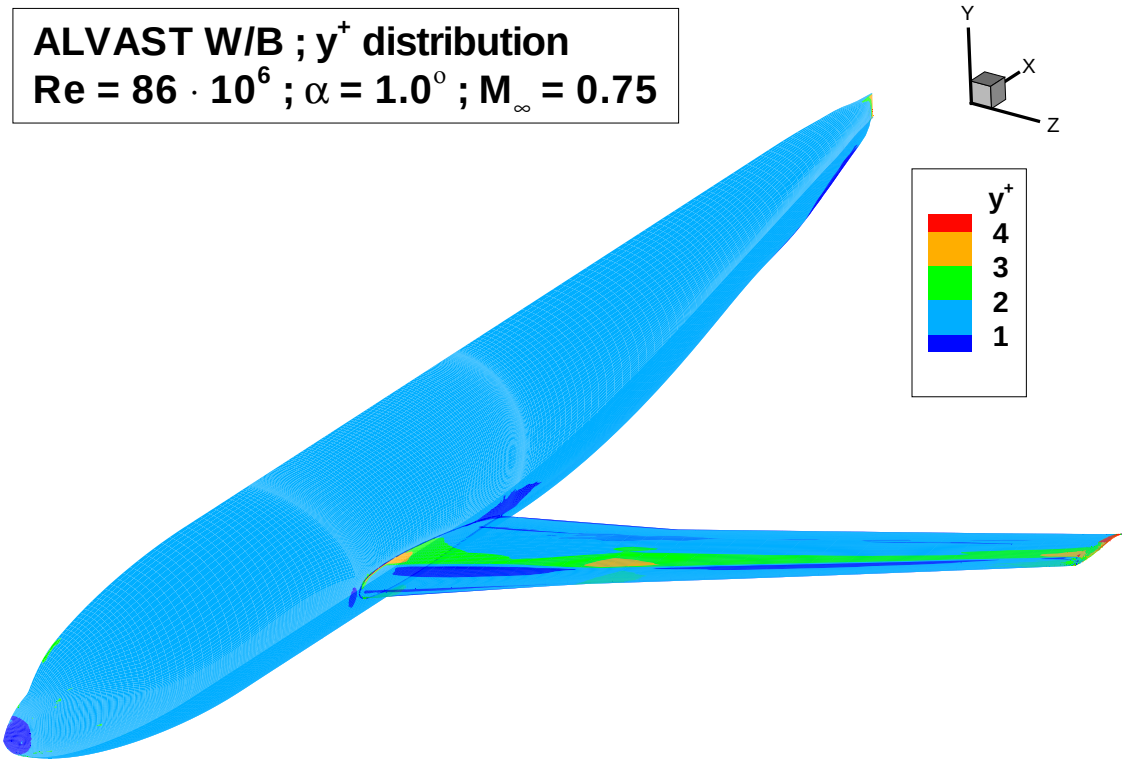

Figure 7.3: y_w^+ distribution; ALVAST Wing/Body configuration, RANS calculation using k–ω turbulence model, Re $= 86 \cdot 10^6$

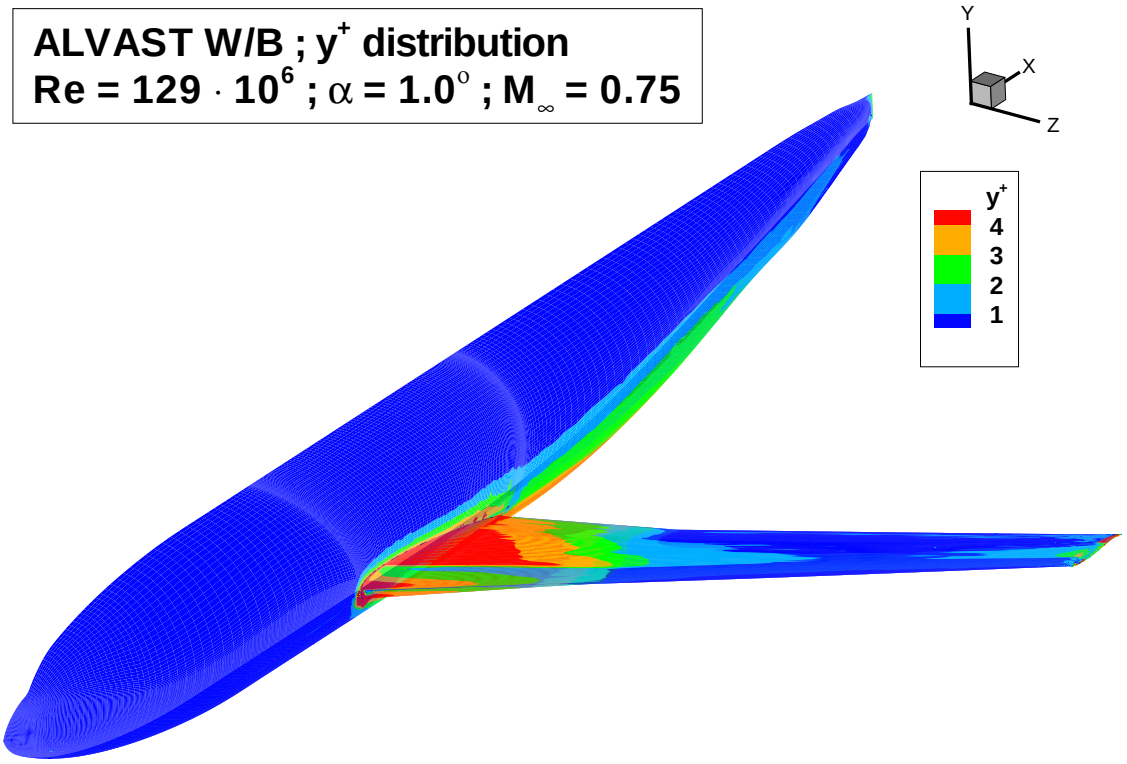

Figure 7.4: y_w^+ distribution; ALVAST Wing/Body configuration, RANS calculation using k–ω turbulence model, Re $= 129 \cdot 10^6$

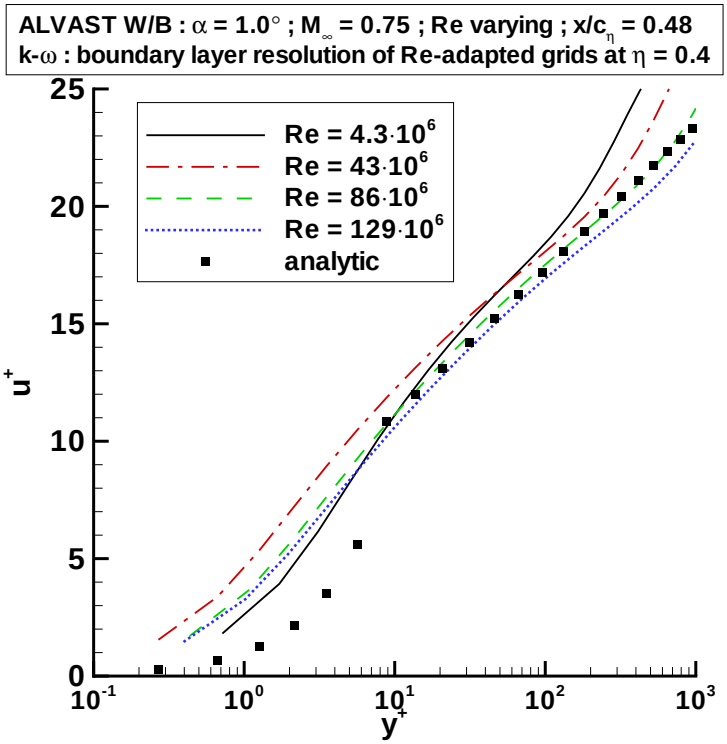

Figure 7.5: Dimensionless velocity profile at $\eta = 0.4$, 48% local chord length (upper side) within boundary layer for Re–adapted grids around ALVAST Wing/Body configuration, varying Reynolds number

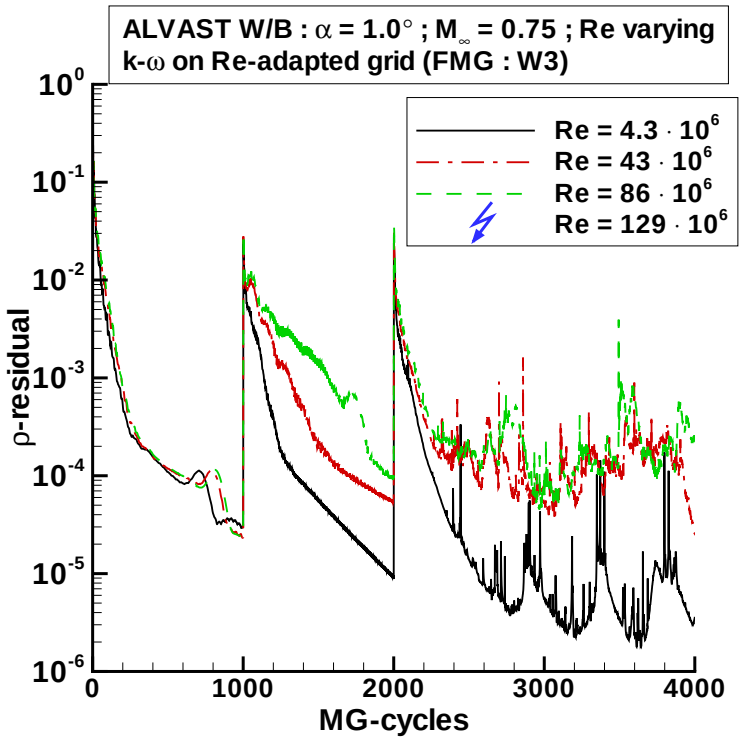

Figure 7.6: Convergence of density residual; ALVAST Wing/Body, RANS calculation using k–ω turbulence model with original FLOWer settings, varying Reynolds number; RANS & k–ω: 3 multigrid level, W–cycle, FMG

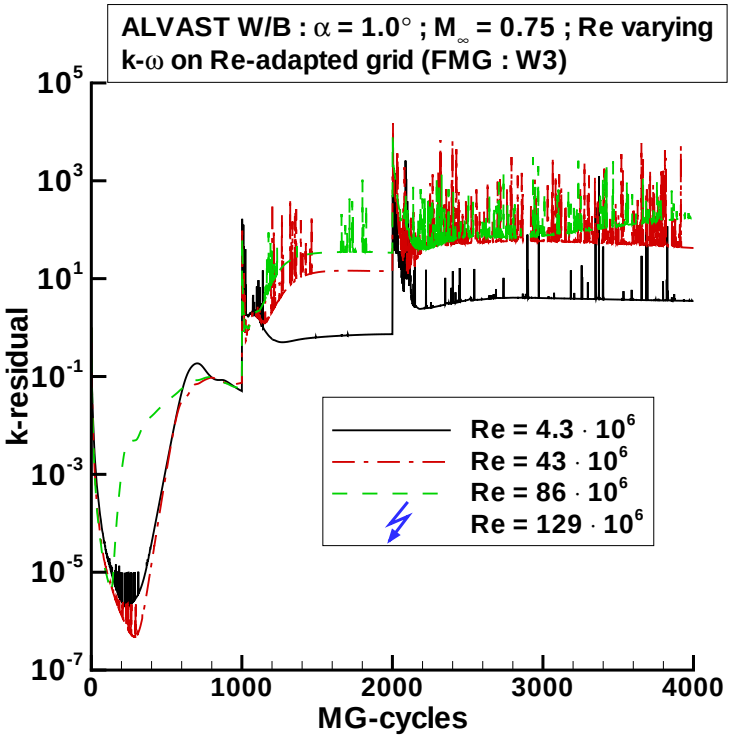

Figure 7.7: Convergence of k residual; ALVAST Wing/Body, RANS calculation using k–ω turbulence model with original FLOWer settings, varying Reynolds number; RANS & k–ω: 3 multigrid level, W–cycle, FMG

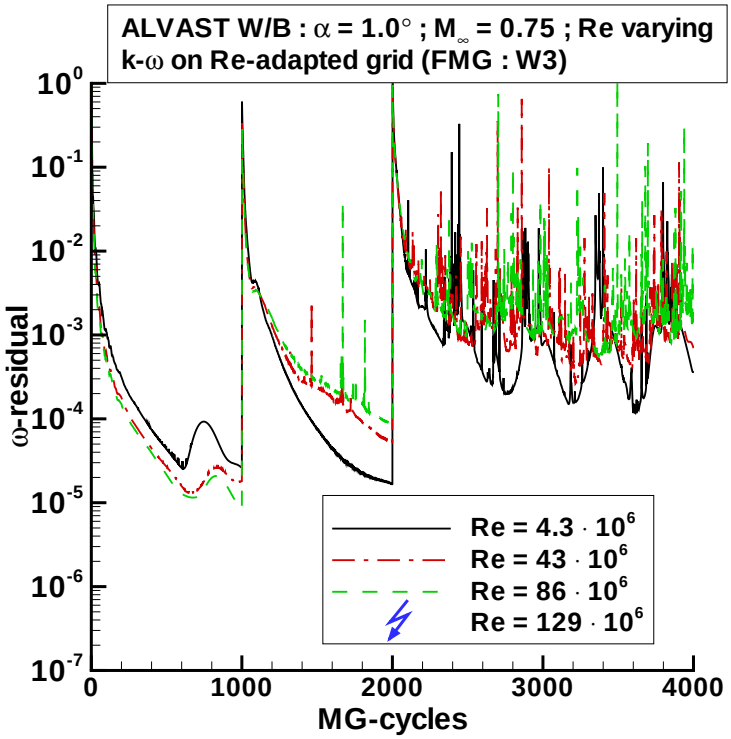

Figure 7.8: Convergence of ω residual; ALVAST Wing/Body, RANS calculation using k–ω turbulence model with original FLOWer settings, varying Reynolds number; RANS & k–ω: 3 multigrid level, W–cycle, FMG

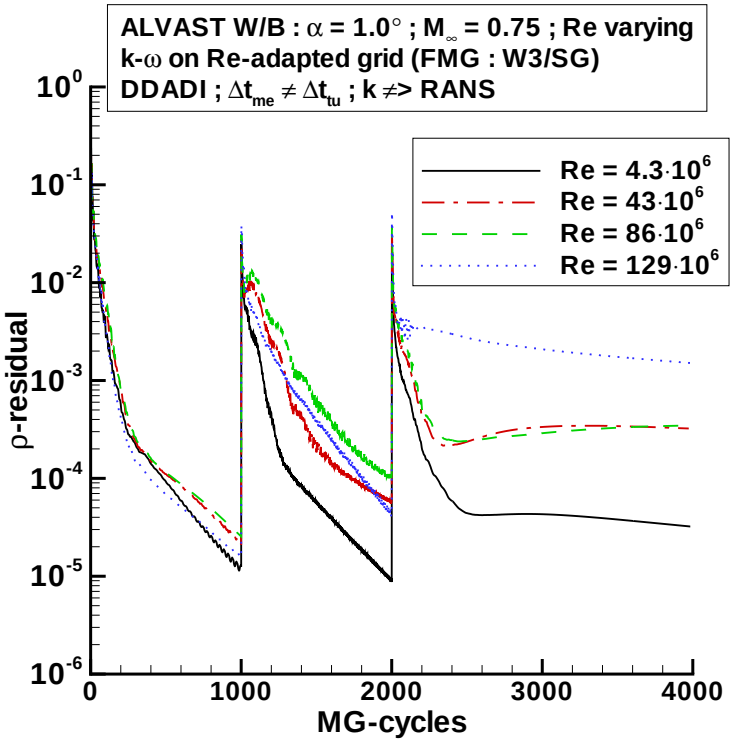

Figure 7.9: Convergence of density residual; ALVAST Wing/Body, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, DDADI, varying Reynolds number; RANS: 3 multigrid level, W–cycle, FMG; k–ω: singlegrid, DDADI, FMG

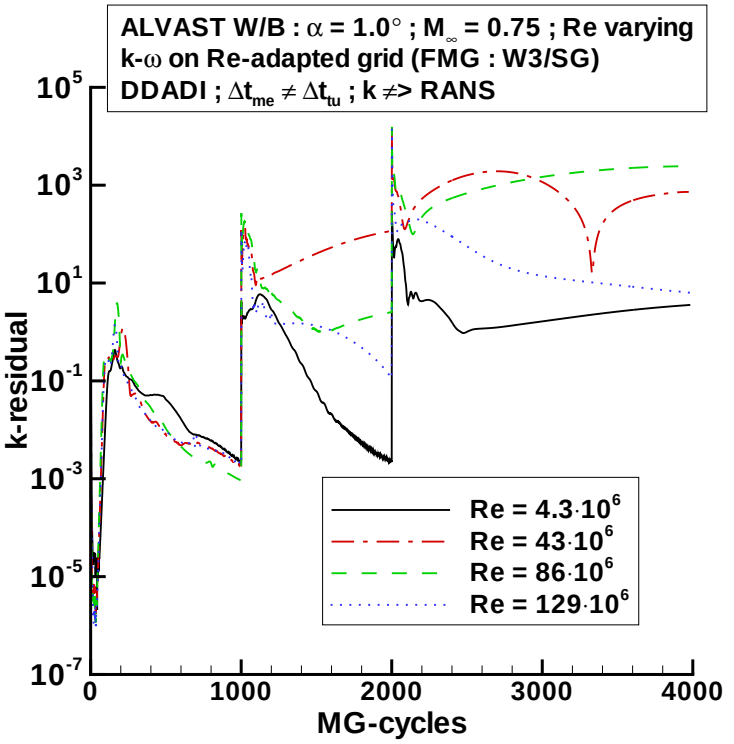

Figure 7.10: Convergence of k residual; ALVAST Wing/Body, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, DDADI, varying Reynolds number; RANS: 3 multigrid level, W-cycle; k–ω: singlegrid

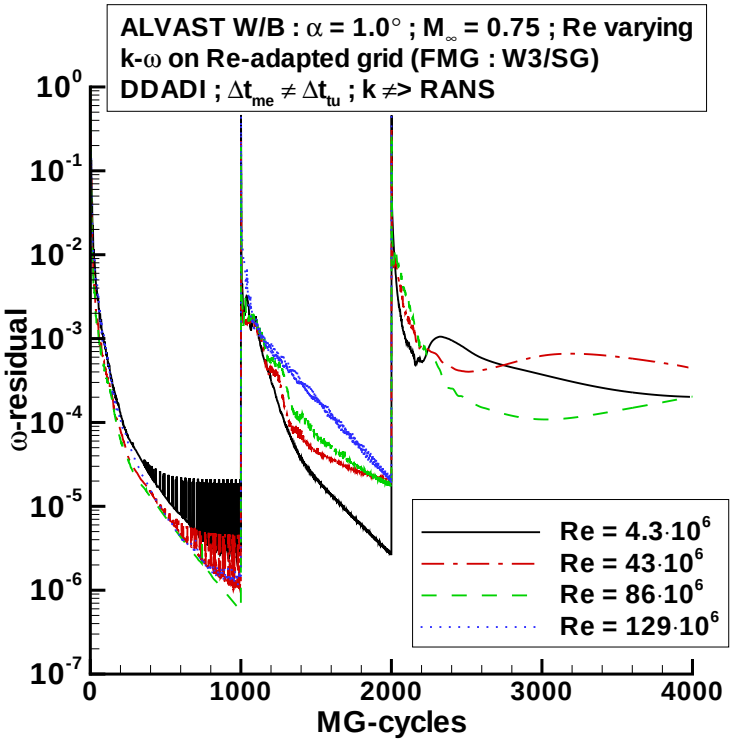

Figure 7.11: Convergence of ω residual; ALVAST Wing/Body, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, DDADI, varying Reynolds number; RANS: 3 multigrid level, W–cycle, FMG; k–ω: singlegrid, DDADI, FMG

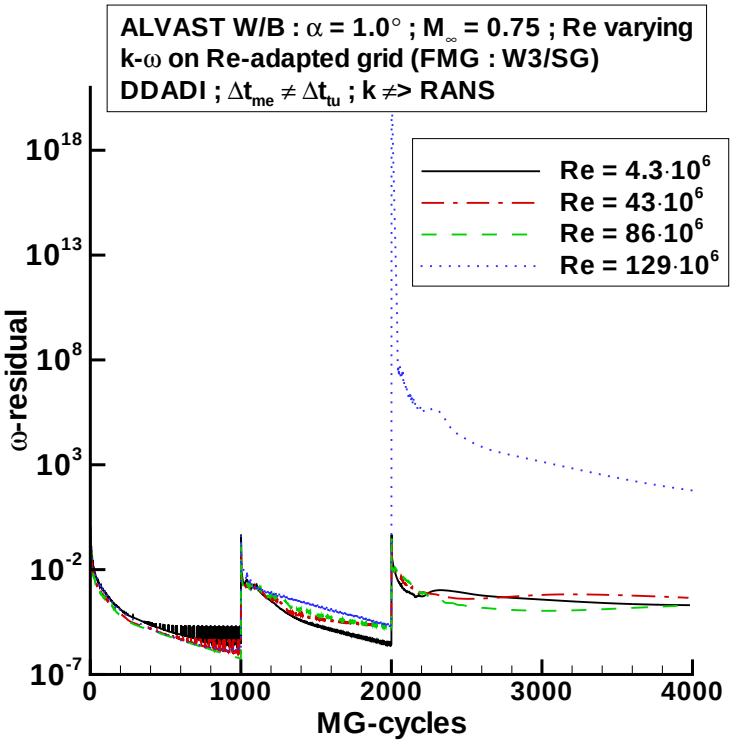

Figure 7.12: Convergence of ω residual; ALVAST Wing/Body, RANS calculation using k–ω turbulence model with modified time step for turbulence equations, DDADI, varying Reynolds number; RANS: 3 multigrid level, W–cycle, FMG; k–ω: singlegrid, DDADI, FMG

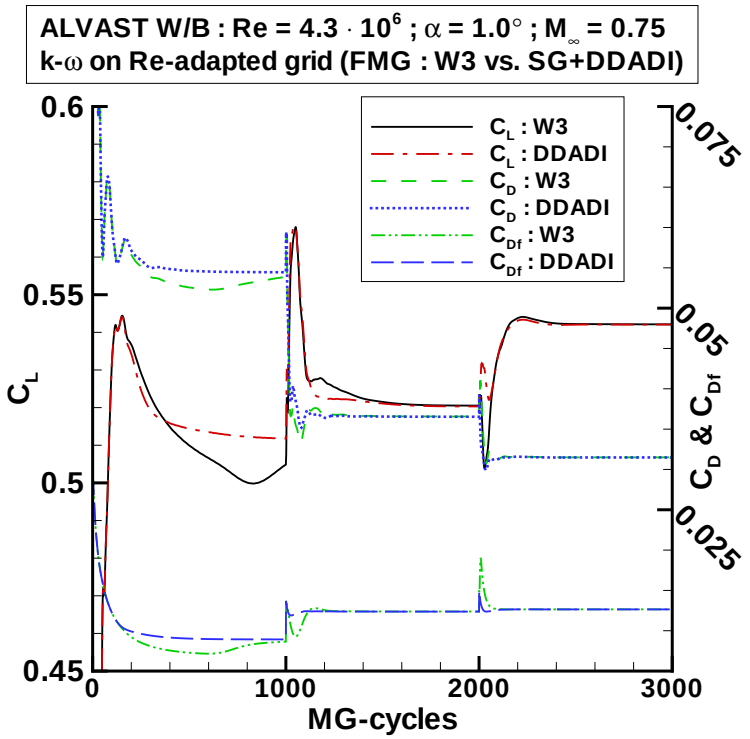

Figure 7.13: Convergence of aerodynamic coefficients; ALVAST Wing/Body, RANS calculation using k–ω turbulence model at Re $= 4.3 \cdot 10^6$; RANS: 3 multigrid level, W–cycle, FMG; k–ω: varying time integration

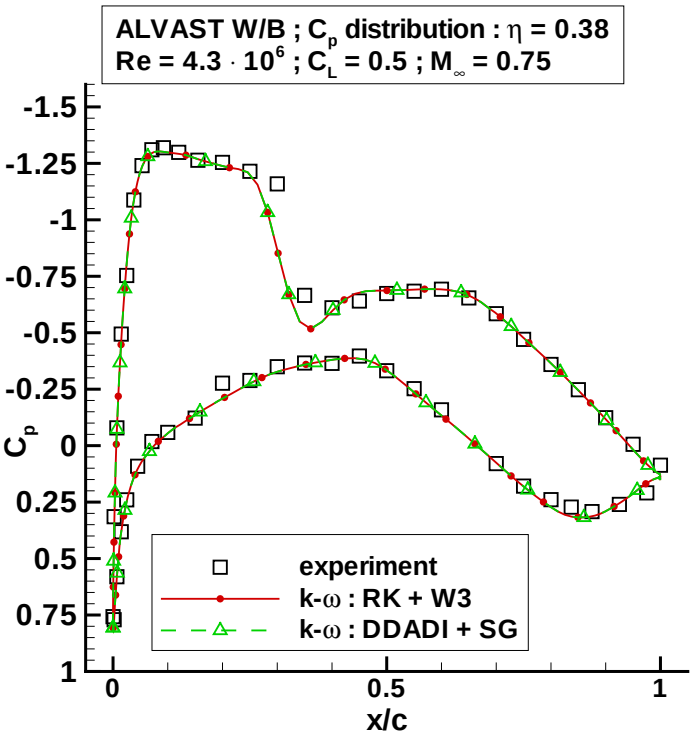

Figure 7.14: C_p distribution at $\eta = 0.38$; ALVAST Wing/Body, RANS calculation using k–ω turbulence model at Re $= 4.3 \cdot 10^6$; RANS: 3 multigrid level, W–cycle, FMG; k–ω: varying time integration

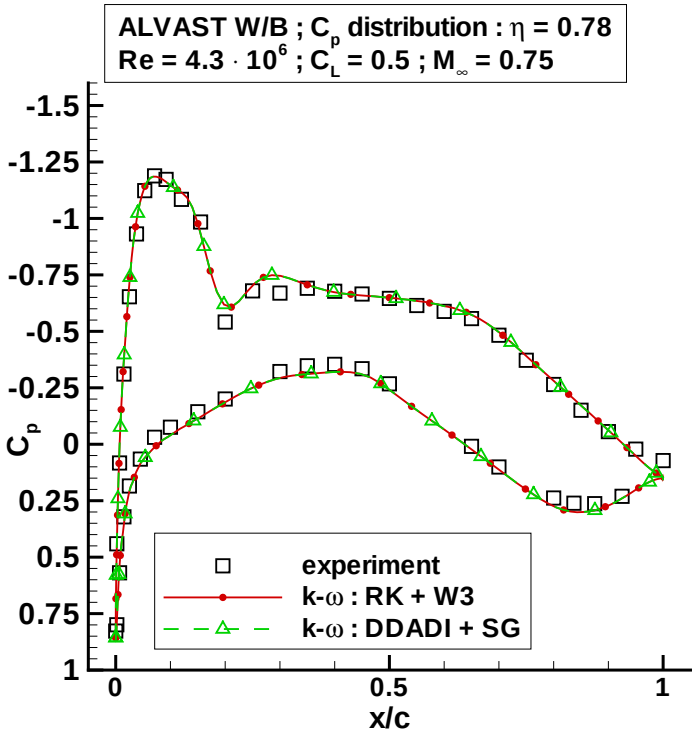

Figure 7.15: C_p distribution at $\eta = 0.78$; ALVAST Wing/Body, RANS calculation using k–ω turbulence model at Re $= 4.3 \cdot 10^6$; RANS: 3 multigrid level, W–cycle, FMG; k–ω: varying time integration

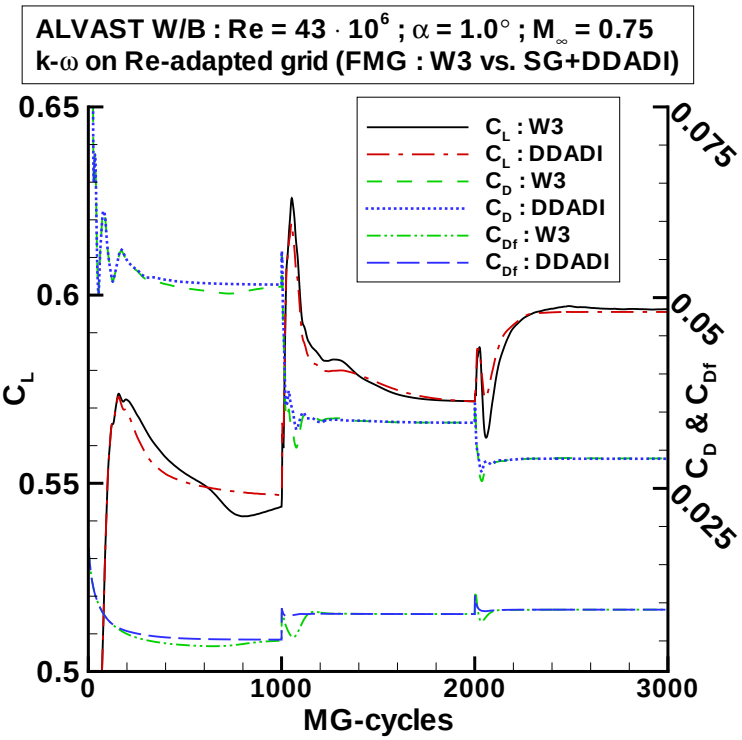

Figure 7.16: Convergence of aerodynamic coefficients; ALVAST Wing/Body, RANS calculation using k–ω turbulence model at Re $= 43 \cdot 10^6$; RANS: 3 multigrid level, W–cycle, FMG; k–ω: varying time integration

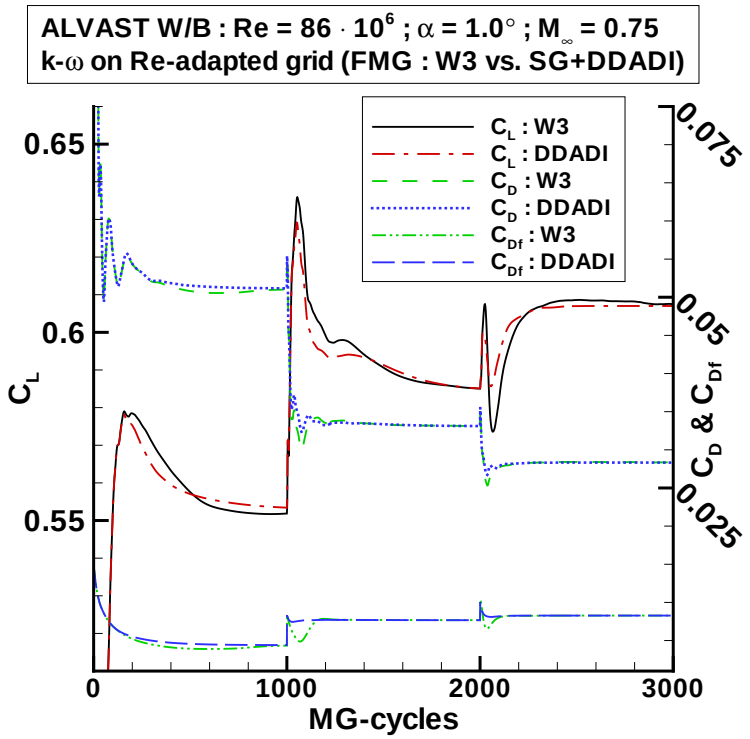

Figure 7.17: Convergence of aerodynamic coefficients; ALVAST Wing/Body, RANS calculation using k–ω turbulence model at Re $= 86 \cdot 10^6$; RANS: 3 multigrid level, W–cycle, FMG; k–ω: varying time integration

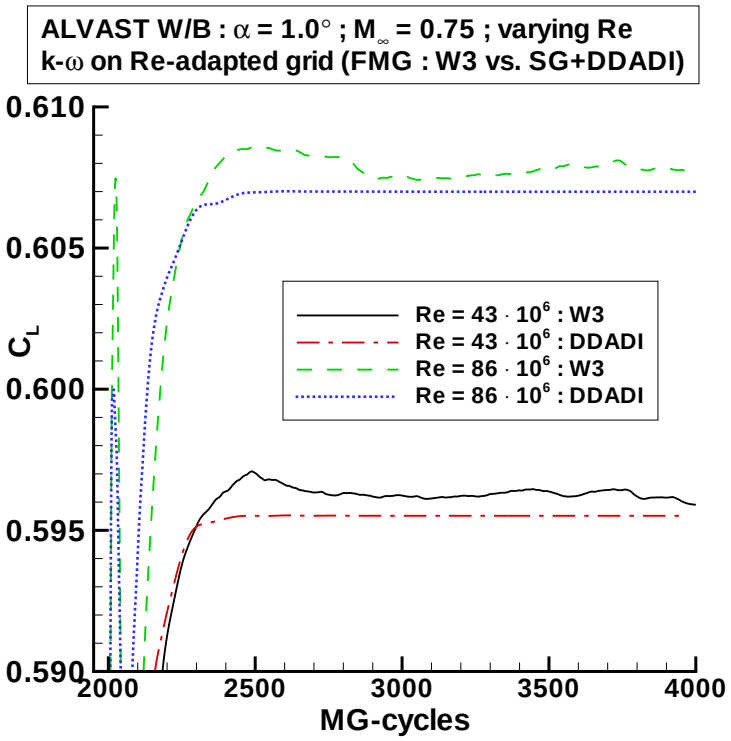

Figure 7.18: Close–up of lift coefficients; ALVAST Wing/Body, RANS calculation using k–ω turbulence model at Re $= 43 \cdot 10^6$ and Re $= 86 \cdot 10^6$; RANS: 3 multigrid level, W–cycle, FMG; k–ω: varying time integration

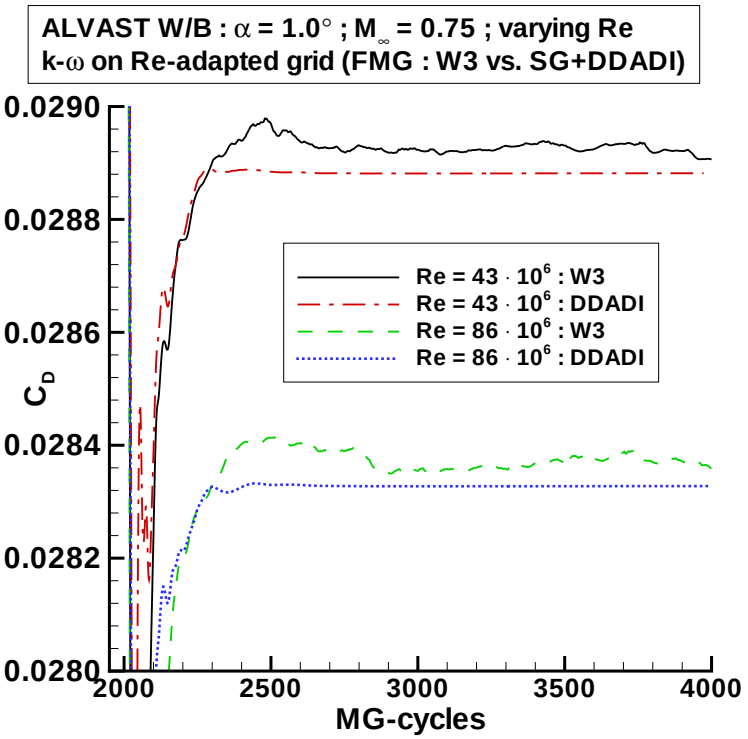

Figure 7.19: Close–up of (total) drag coefficients; ALVAST Wing/Body, RANS calculation using k–ω turbulence model at Re $= 43 \cdot 10^6$ and Re $= 86 \cdot 10^6$; RANS: 3 multigrid level, W–cycle, FMG; k–ω: varying time integration

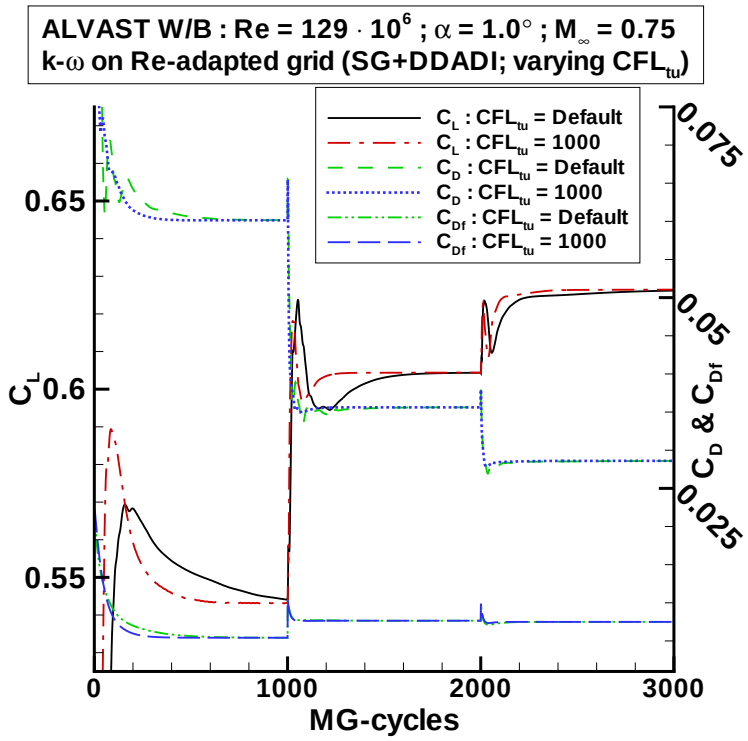

Figure 7.20: Convergence of aerodynamic coefficients; ALVAST Wing/Body, RANS calculation using k–ω turbulence model at Re $= 129 \cdot 10^6$; RANS: 3 multigrid level, W–cycle, FMG; k–ω: singlegrid, DDADI, FMG, varying CFLTU

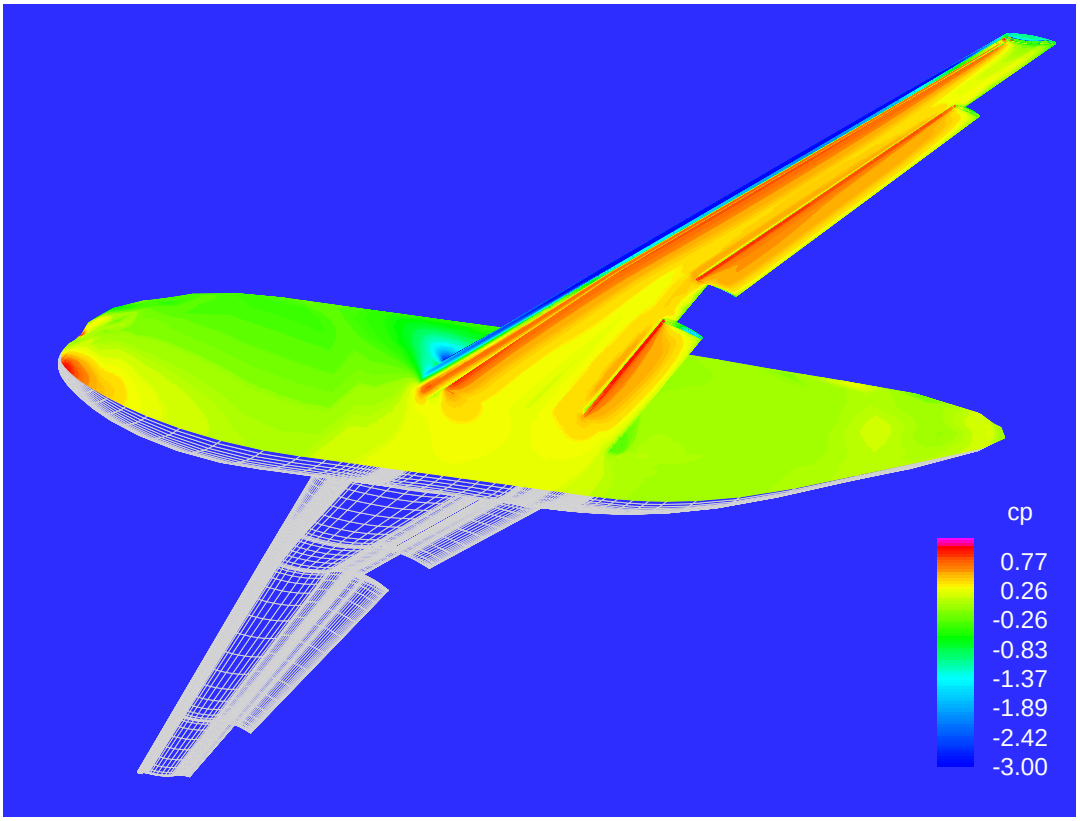

Figure 7.21: Computational surface grid and pressure distribution for ALVAST Wing/Body High–Lift configuration, RANS calculation using k–ω turbulence model (Figure by courtesy of J. Raddatz, DLR)

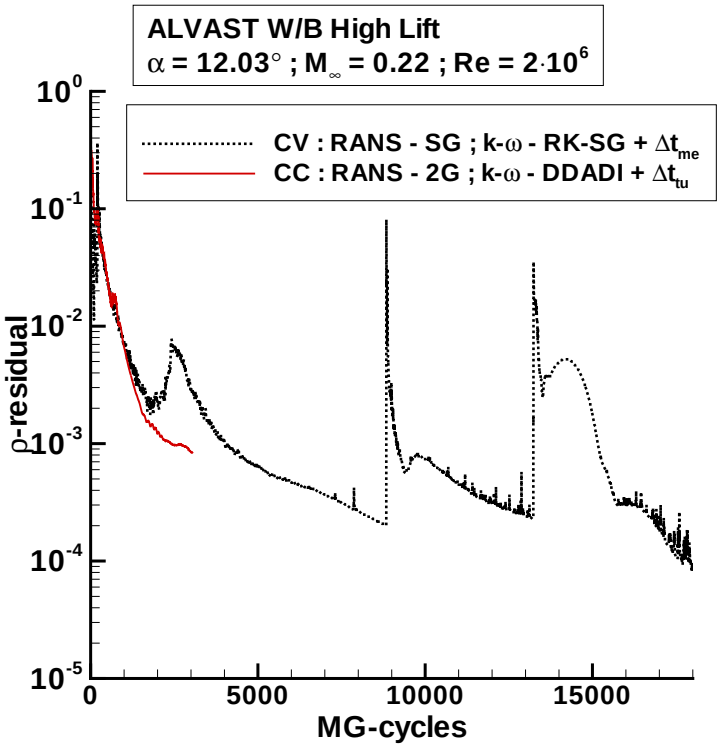

Figure 7.22: Convergence of density residual; ALVAST Wing/Body High–Lift configuration, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations (Data by courtesy of J. Raddatz, DLR)

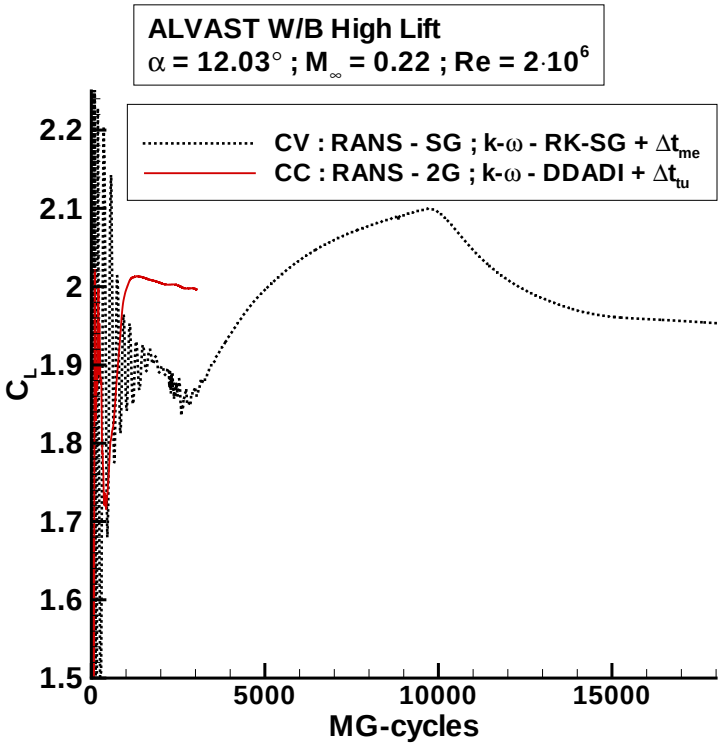

Figure 7.23: Convergence of lift coefficient; ALVAST Wing/Body High–Lift configuration, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations (Data by courtesy of J. Raddatz, DLR)

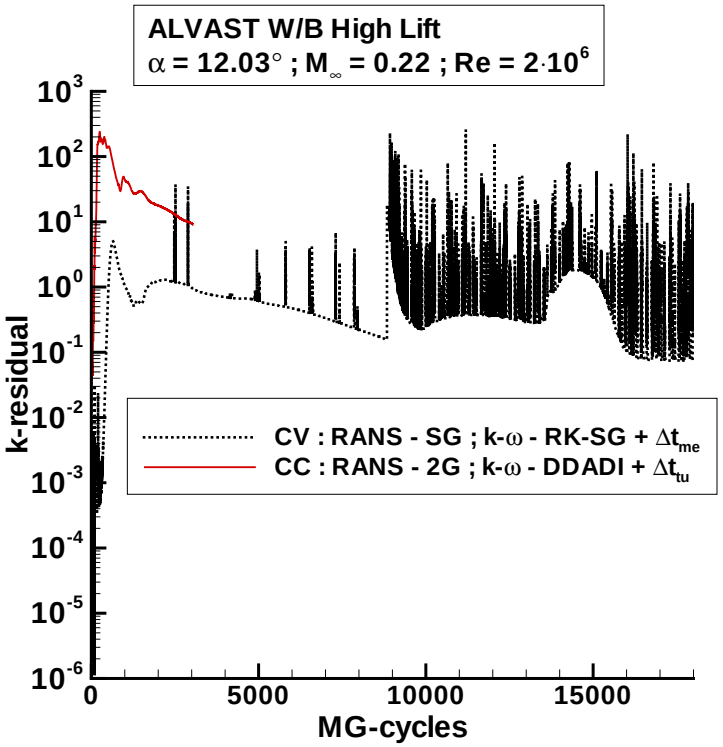

Figure 7.24: Convergence of k residual; ALVAST Wing/Body High–Lift configuration, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations (Data by courtesy of J. Raddatz, DLR)

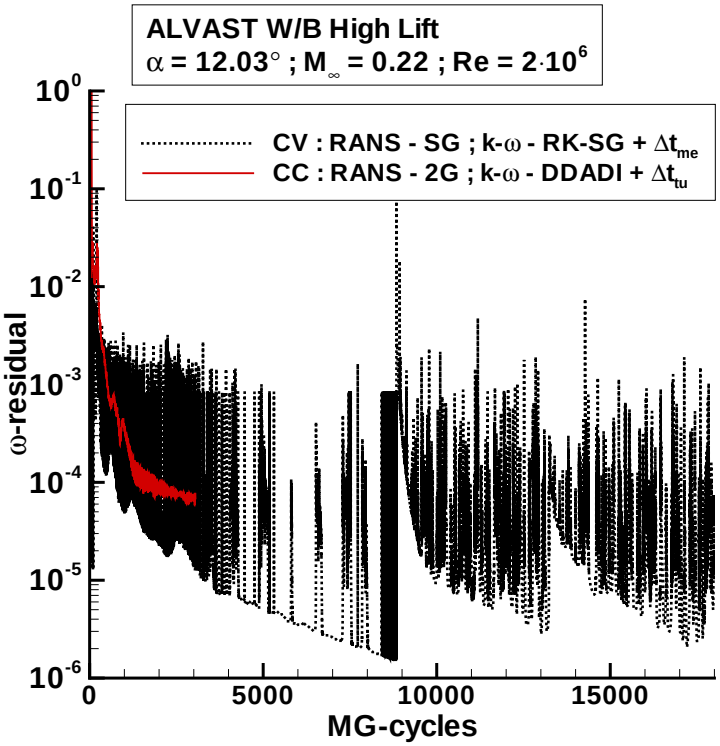

Figure 7.25: Convergence of ω residual; ALVAST Wing/Body High–Lift configuration, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations (Data by courtesy of J. Raddatz, DLR)

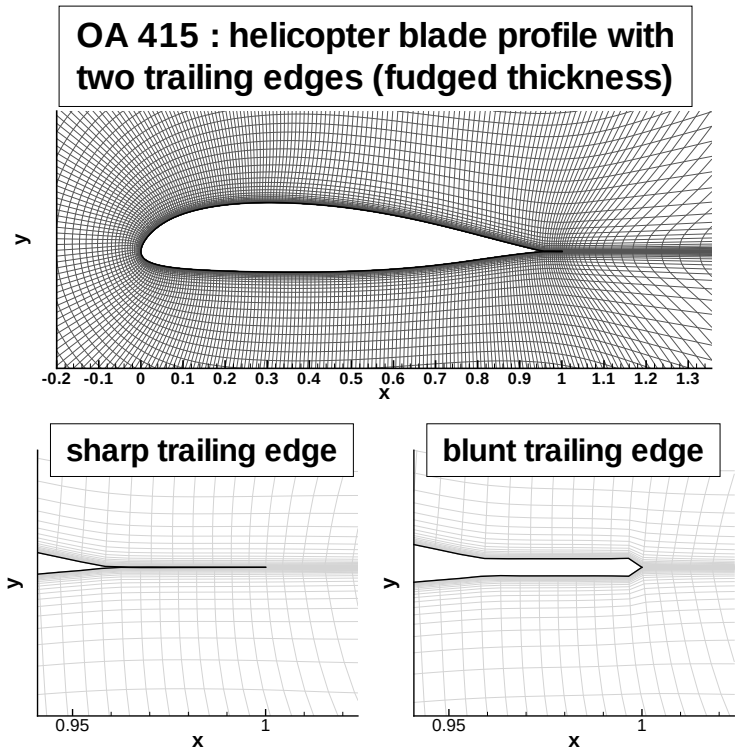

Figure 7.26: Computational grid for OA415 airfoil with geometry details of sharp and blunt trailing edge

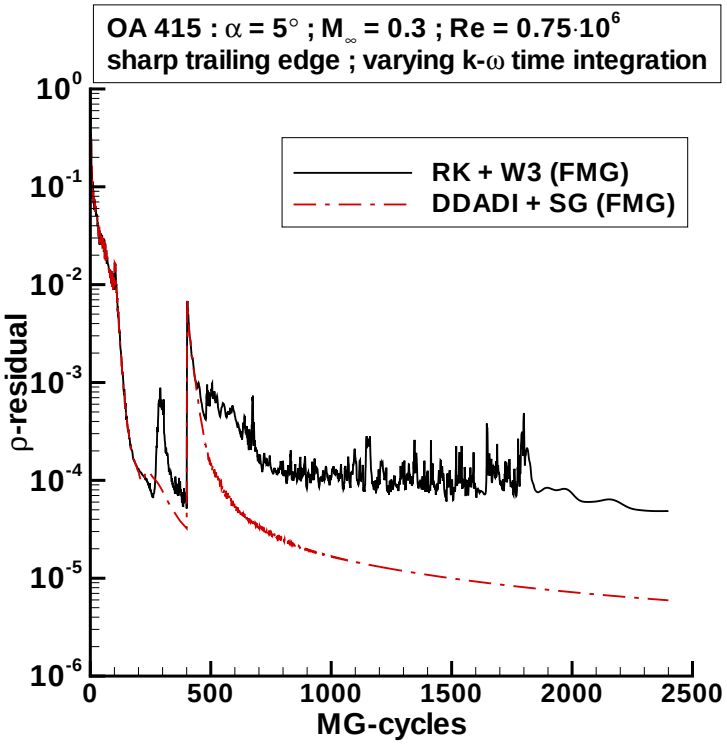

Figure 7.27: Convergence of density residual; OA415 airfoil with sharp trailing edge, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations; RANS : 3 multigrid level, W–cycle, FMG

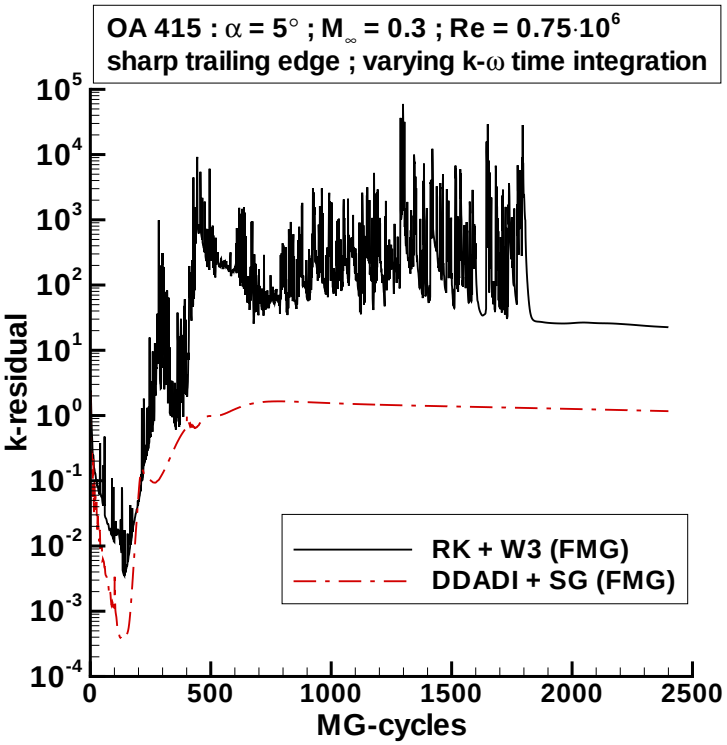

Figure 7.28: Convergence of k residual; OA415 airfoil with sharp trailing edge, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations; RANS : 3 multigrid level, W–cycle, FMG

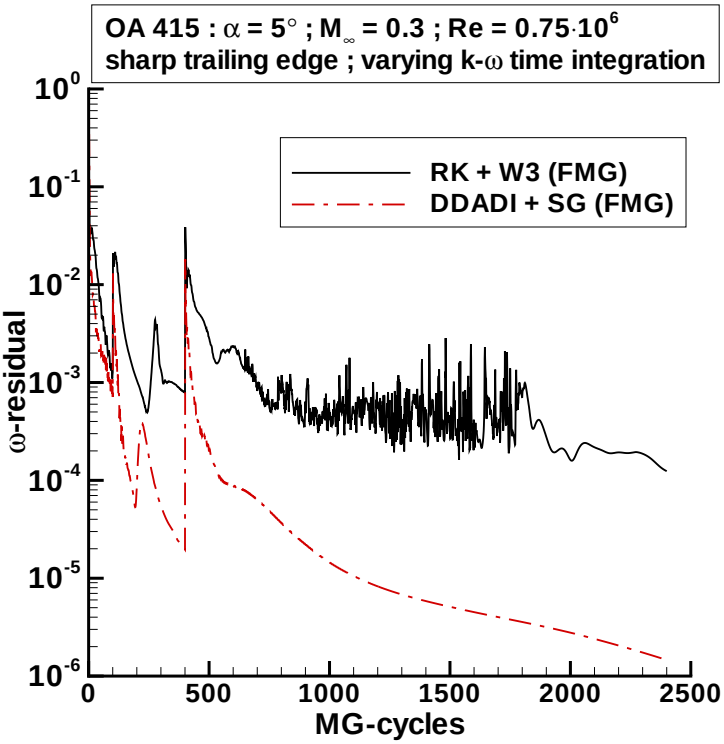

Figure 7.29: Convergence of ω residual; OA415 airfoil with sharp trailing edge, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations; RANS : 3 multigrid level, W–cycle, FMG

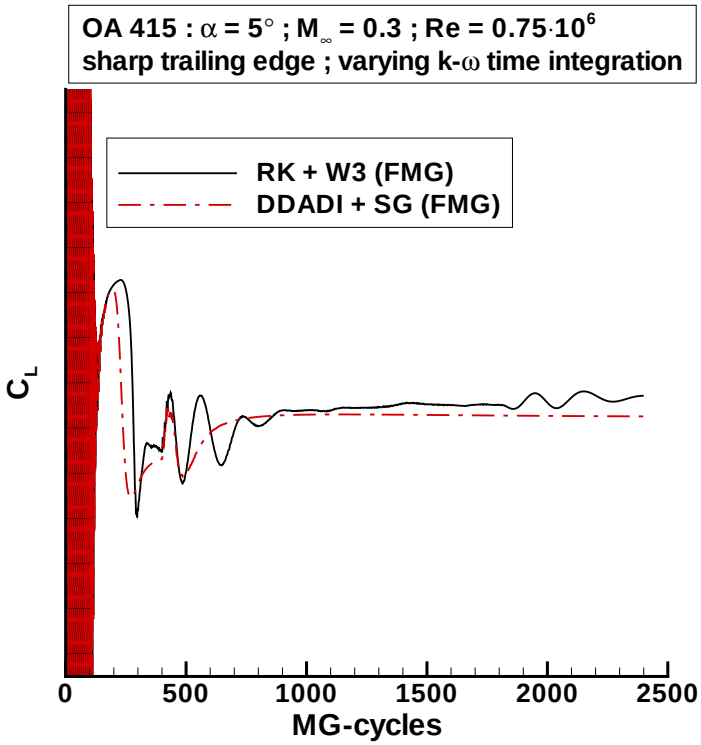

Figure 7.30: Convergence of lift coefficient; OA415 airfoil with sharp trailing edge, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations; RANS : 3 multigrid level, W–cycle, FMG

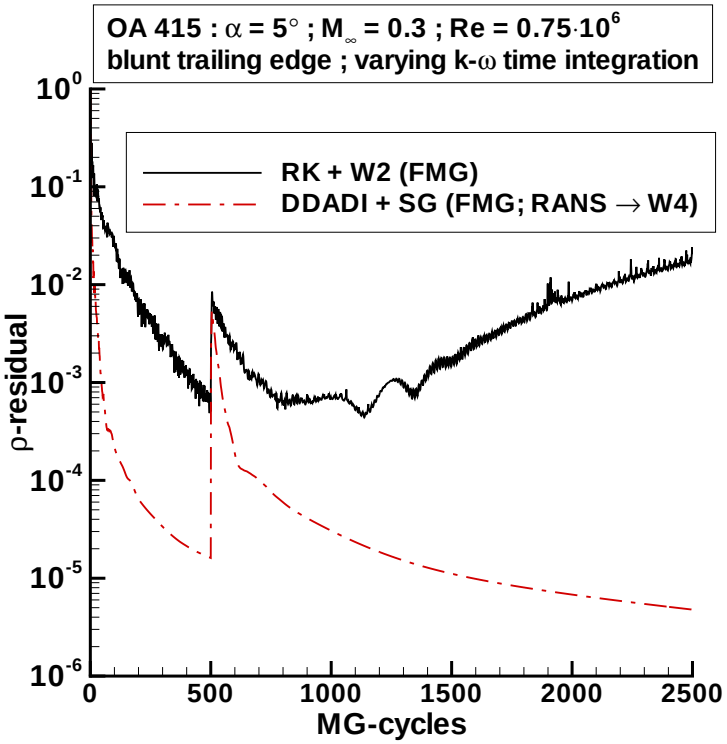

Figure 7.31: Convergence of density residual; OA415 airfoil with blunt trailing edge, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations; RANS : 2 resp. 4 multigrid level, W–cycle, FMG

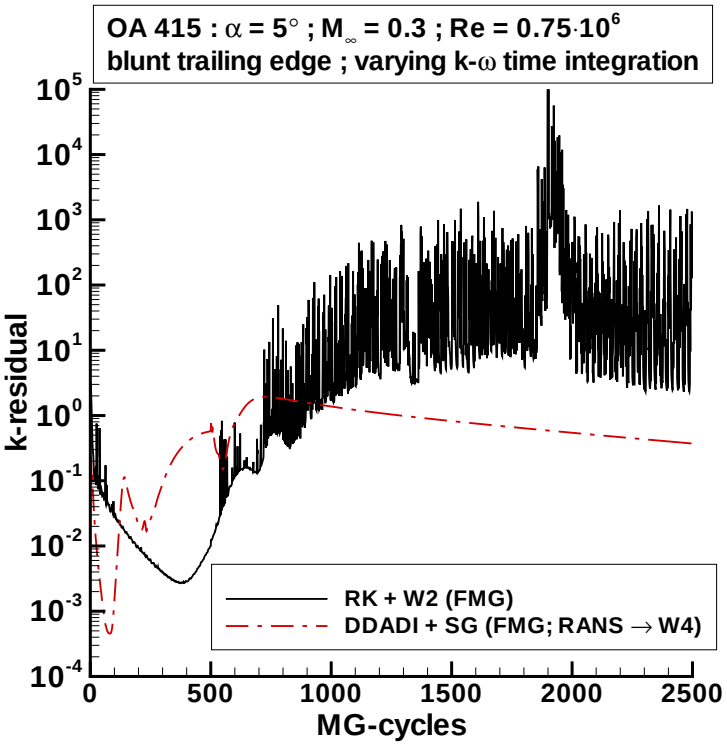

Figure 7.32: Convergence of k residual; OA415 airfoil with blunt trailing edge, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations; RANS : 2 resp. 4 multigrid level, W–cycle, FMG

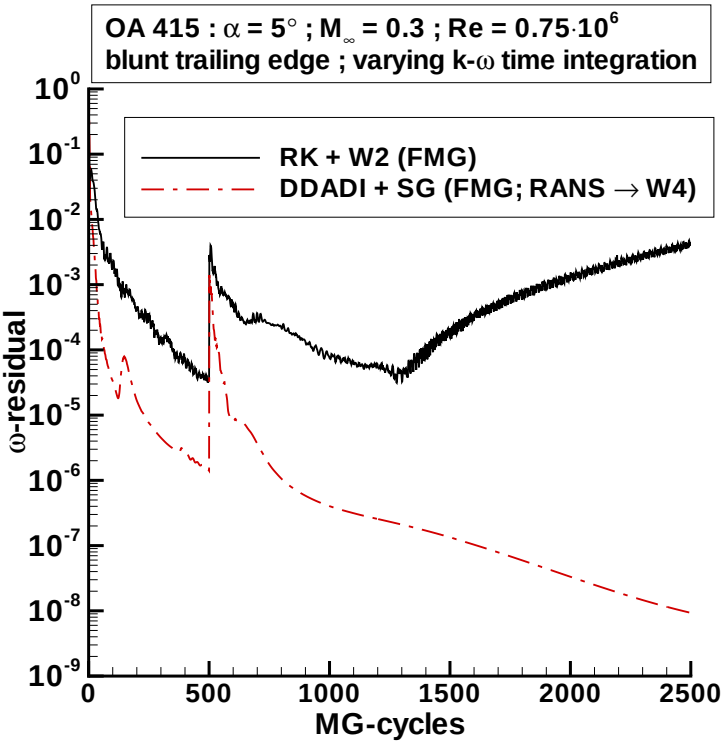

Figure 7.33: Convergence of ω residual; OA415 airfoil with blunt trailing edge, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations; RANS : 2 resp. 4 multigrid level, W–cycle, FMG

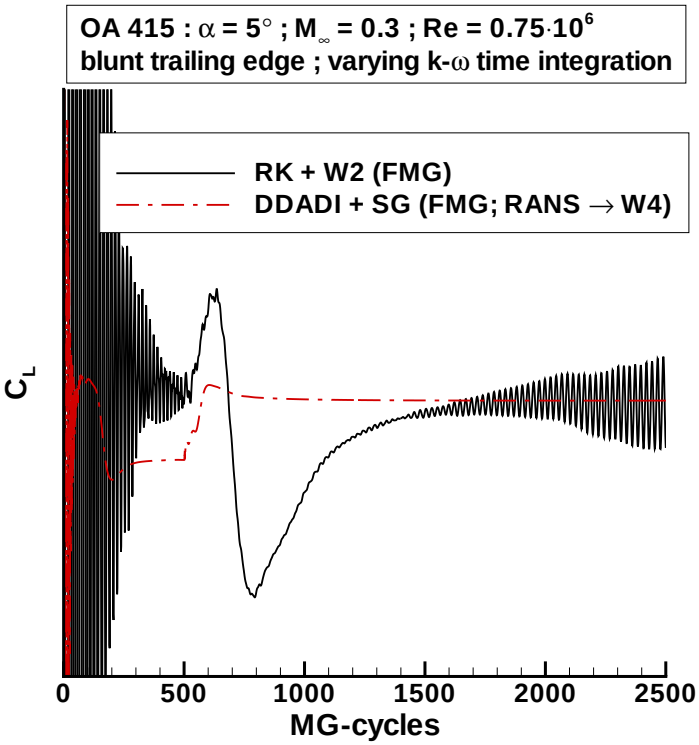

Figure 7.34: Convergence of lift coefficient; OA415 airfoil with blunt trailing edge, RANS calculation using k–ω turbulence model with varying time integration scheme for k–ω equations; RANS : 2 resp. 4 multigrid level, W–cycle, FMG

Bibliography

[1] J. D. Anderson. *Computational Fluid Dynamics: The Basics with Applications*. McGraw Hill, New York, 1995.

[2] U. Ascher, R. M. M. Mattheij, and R. D. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[3] P. Aumann, H. Barnewitz, H. Schwarten, K. Becker, R. Heinrich, B. Roll, M. Galle, N. Kroll, T. Gerhold, D. Schwamborn, and M. Franke. MEGAFLOW: Parallel complete aircraft CFD. *Parallel Computing*, 27:415–440, 2001.

[4] B. S. Baldwin and H. Lomax. Thin layer approximation and algebraic model for separated turbulent flows, 1978. *AIAA paper* 78–0257.

[5] J. Bardina and C. K. Lombard. Three dimensional hypersonic flow simulations with the CSCM implicit upwind Navier–Stokes method, 1987. *AIAA paper* 87–1114.

[6] J. Blazek. *Verfahren zur Beschleunigung der Lösung der Euler– und Navier–Stokes–Gleichungen bei stationären Über- und Hyperschallströmungen (Methods to Accelerate the Solution of the Euler– and the Navier–Stokes–Equations for Steady–State Super- and Hypersonic Flows)*. Doctoral thesis, Technische Universität Braunschweig, 1994. (Published as DLR research report DLR–FB 94–35).

[7] J. Blazek. *Computational Fluid Dynamics: Principals and Applications*. Elsevier, Amsterdam, 2001.

[8] A. I. Borisenko and I. E. Tarapov. *Vector and Tensor Analysis with Applications*. Dover Publications, Inc., New York, 1968.

[9] J. Boussinesq. *Essai sur la théorie des eaux courantes*. Imprimerie Nationale, Paris, 1877.

[10] A. Brandt. Guide to multigrid development. In Hackbusch and Trottenberg [43], pages 220–312.

[11] A. Brandt. Rigorous quantative analysis of multigrid: I. constant coefficients two level cycle with l_2 norm. *SIAM Journal on Numerical Analysis*, (31):1695–1730, 1994.

[12] A. Brandt. *Barriers to Achieving Textbook Multigrid Efficiency (TME) in CFD*. ICASE Interim Report 32, April 1998.

[13] A. Brandt and I. Livshits. Wave–ray multigrid method for standing wave equations. *Electronic Transactions on Numerical Analysis*, 6:162–181, 1997.

[14] W. L. Briggs. *A Multigrid Tutorial*. SIAM — Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1987.

[15] W. R. Briley and H. McDonald. An overview and generalization of implicit Navier–Stokes algorithms and approximate factorization. *Computers & Fluids*, 30:807–828, 2001.

[16] O. Brodersen, M. Hepperle, A. Ronzheimer, C.-C. Rossow, and B. Schöning. *Entwicklung des parametrischen Netzgenerierungssystems MegaCads, Statusbericht 1-96*. DLR–IB 129–96/1, January 1996.

[17] O. Brodersen, M. Hepperle, A. Ronzheimer, C.-C. Rossow, and B. Schöning. The parametric grid generation system MegaCads. In B. K. Soni, J. F. Thompson, J. Häuser, and P. Eisemann, editors, *Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Field Simulations, 1.-5.4.1996, NSF Engineering Research Center, Mississippi State University*, pages 353–362, 1996.

[18] O. Brodersen, E. Monsen, A. Ronzheimer, R. Rudnik, and C.-C. Rossow. Computation of aerodynamic coefficients for the DLR–F6 configuration using MEGAFLOW. In Nitsche et al. [90], pages 85–92.

[19] P. E. O. Buelow, S. Venkateswaran, and C. L. Merkle. The effect of grid aspect ratio on convergence, 1993. *AIAA paper* 93–3367–CP.

[20] W. Burgsmüller and H. Hoheisel. ENIFAIR — EU research into engine integration on future transport aircraft. *Air & Space Europe*, 2(2):81–85, 2000.

[21] P. Catalano and M. Amato. Assessment of k–ω turbulence modelling in the CIRA flow solver ZEN. In *ECCOMAS 2001*, 2001. Proceedings–CD–ROM.

[22] T. Cebeci and A. M. O. Smith. *Analysis of Turbulent Boundary Layers*. Academic Press, New York, 1974.

[23] F. Le Chuiton. *Une méthode implicite non-factorisée décentrée hybride pour la simulation numérique d'écoulements autour de géométries complexes (A Hybrid Upwind Non–Factorized Implicit Method for the Numerical Simulation of Flows Around Complex Geometries)*. Doctoral thesis, École Centrale de Lyon, 1999.

[24] R. Collercandy. Multigrid strategy for Euler and Navier–Stokes computations for flows around complex aerospace configurations. In *ECCOMAS '98*, pages 1–6, 1998.

[25] P. H. Cook, M. A. McDonald, and M. C. P. Firmin. *Aerofoil RAE 2822 — Pressure Distributions and Boundary Layer and Wake Measurements*, chapter 6. Number 138 in AGARD–AR. AGARD — Advisory Group for Aerospace Research & Development, Neuilly-sur-Seine, France, 1979.

[26] A. Corjon, D. Darracq, S. Champagneux, C. Gacherieu, and T. Schönfeld. Simulation of aircraft in high–lift configuration, 1999. *AIAA paper* 99–3179.

[27] V. Couaillier, P. Eliasson, and J. Faßbender. Enhancing robustness for advanced turbulence models in the European project AVTAC, 2000. *AIAA paper* 2000–2407.

[28] E. Dick and J. Steelant. Coupled solution of the steady compressible Navier–Stokes equations and the k–ε turbulence equations with a multigrid method. *Applied Numerical Mathematics*, 23:49–61, 1997.

[29] DLR, Institute of Design Aerodynamics. *FLOWer — Installation and User Handbook, Release 116*, May 2000.

[30] D. Drikakis, O. P. Iliev, and D. P. Vassileva. A nonlinear multigrid method for the three–dimensional incompressible Navier–Stokes equations. *Journal of Computational Physics*, 146:301–321, 1998.

[31] L. Dubuc, K. J. Badcock, B. E. Richards, and M. Woodgate. Implicit Navier–Stokes simultations of unsteady flows. In *Unsteady Aerodynamics — Two Day Conference, 17-18 July 1996, London, UK*, pages 11.1–11.9, London, 1996. Royal Aeronautical Society.

[32] J. R. Edwards. An implicit multigrid algorithm for computing hypersonic, chemically reacting viscous flows. *Journal of Computational Physics*, 123:84–95, 1996.

[33] B. Eisfeld. *Die Reynolds–Spannungsgleichungen für kompressible Strömung — Herleitung und Zusammenhänge* . DLR–IB 124–2002/36, October 2002.

[34] B. Eisfeld and R. Rudnik. *Der Einfluss der Schließungskoeffizienten und der ω–Wandrandbedingung auf die Lösung der RANS–Gleichungen mit dem k–ω–Turbulenzmodell von Wilcox bei transsonischer Strömung*. DLR–IB 124–2002/35, October 2002.

[35] H. C. Elman, O. G. Ernst, and D. P. O'Leary. A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations, June 1999. *http://www.mgnet.org/mgnet-papers.html#E*.

[36] R. Emunds, Airbus Deutschland GmbH. private communications.

[37] J. Fassbender. Experiences with multigrid–prolongation for two–equation turbulence models in flows with high Reynolds numbers. In E. Dick, K. Riemslagh, and J. Vierendeels, editors, *Multigrid Methods VI — Proceedings, Gent, 1999*, volume 14 of *Lecture Notes in Computational Science and Engineering*, pages 108–114, Berlin, 2000. Springer–Verlag.

[38] M. Galle. *Ein Verfahren zur numerischen Simulation kompressibler reibungsbehafteter Strömungen auf hybriden Netzen (A Method for the Numerical Simulation of Compressible Viscous Flows on Hybrid Grids)*. Doctoral thesis, Universität Stuttgart, 1998. (Published as DLR research report DLR–FB 99–04).

[39] P. Gerlinger, H. Möbus, and D. Brüggemann. An implicit multigrid method for turbulent combustion. *Journal of Computational Physics*, 167:247–276, 2001.

[40] P. Gerlinger, P. Stoll, and D. Brüggemann. An implicit multigrid method for the simulation of chemically reacting flows. *Journal of Computational Physics*, 146:322–345, 1998.

[41] G. A. Gerolymos. Implicit multiple–grid solution of the compressible Navier–Stokes equations using k–ϵ turbulence closure. *AIAA Journal*, 28(10):1707–1717, 1990.

[42] A. Gould, J.-C. Courty, M. Sillen, E. Elsholz, and A. Abbas. The AVTAC project — a review of European aerospace CFD. In *ECCOMAS 2000*, pages 1–17, 2000.

[43] W. Hackbusch and U. Trottenberg, editors. *Multigrid Methods — Proceedings, Köln, 1981*, volume 960 of *Lecture Notes in Mathematics*, Berlin, 1982. Springer–Verlag.

[44] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Number 8 in Springer Series in Computational Mathematics. Springer–Verlag, Berlin, 2nd edition, 1993.

[45] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential Algebraic Problems*. Number 14 in Springer Series in Computational Mathematics. Springer–Verlag, Berlin, 1991.

[46] C. Hirsch. *Numerical Computations of Internal and External Flows*, volume 1. J. Wiley & Sons, Chichester, 1988.

[47] C. Hirsch. *Numerical Computations of Internal and External Flows*, volume 2. J. Wiley & Sons, Chichester, 1990.

[48] S. O. Ibraheem and A. O. Demuren. *On Bi–Grid Local Mode Analysis of Solution Techniques for 3–D Euler and Navier–Stokes Equations*. ICOMP–94–6 (NASA Technical Memorandum 106749, NASA Lewis), October 1994.

[49] A. Jameson. *Numerical Solution of the Euler Equation for Compressible Inviscid Flow*. Report MAE 1643, Princeton University.

[50] A. Jameson. The evolution of computational methods in aerodynamics. *Journal of Applied Mechanics*, 50:1052–1070, 1983.

[51] A. Jameson. Multigrid algorithms for compressible flow calculations. In W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods II — Proceedings, Köln, 1985*, volume 1228 of *Lecture Notes in Mathematics*, pages 166–201, Berlin, 1986. Springer–Verlag.

[52] A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings, 1991. *AIAA paper* 91–1596.

[53] A. Jameson. *Essential Elements of Computational Algorithms for Aerodynamic Analysis and Design*. ICASE Report 97–68, December 1997.

[54] A. Jameson and T. J. Baker. Solution of the Euler equations for complex configurations. In *6th Computational Fluid Dynamics Conference, July, 1983/Danvers, MA*, pages 293–302, 1983. (*AIAA paper* 83–1929).

[55] A. Jameson and T. J. Baker. Multigrid solution of the Euler equations for aircraft configurations, 1984. *AIAA paper* 84–0093.

[56] A. Jameson, W. Schmidt, and E. Turkel. Numerical solutions of the Euler equations by finite volume methods using Runge–Kutta time–stepping schemes, 1981. *AIAA paper* 81–1259.

[57] W. Joppich. *Grundlagen der Mehrgittermethode — Eine Einführung in Standardverfahren*. Skriptum zur Vorlesung Algorithmen I und II, Fachhochschule Köln, Abteilung Gummersbach, February 1996.

[58] R. Kiock. *The ALVAST Modell of DLR*. DLR–IB 129–96/22, October 1996.

[59] J. C. Kok and S. P. Spekreijse. *Efficient and accurate implementation of the k–ω turbulence model in the NLR multi–block Navier–Stokes system*. NLR–TP–2000–144, May 2000. (based on a presentation held at ECCOMAS 2000, Barcelona, Spain).

[60] S. L. Krist, R. T. Biedron, and C. L. Rumsey. *CFL3D User's Manual (Version 5.0)*, second edition, September 1997.
http://cfl3d.larc.nasa.gov/cfl3dv6/cfl3dv6.html.

[61] N. Kroll. National CFD project MEGAFLOW — status report —. In H. Körner and R. Hilbig, editors, *New Results in Numerical and Experimental Fluid Mechanics — Contributions to the 10th AG STAB/DGLR Symposium Braunschweig, Germany 1996*, volume 60 of *Notes on Numerical Fluid Mechanics*, pages 15–23, Braunschweig, 1999. Vieweg Verlag.

[62] N. Kroll, B. Eisfeld, and H. M. Bleecke. FLOWer. In *Portable Parallelization of Industrial Aerodynamic Applications (POPINDA)*. Vieweg Verlag, Braunschweig, 1999.

[63] N. Kroll and R. K. Jain. *Solution of Two–Dimensional Euler Equations — Experience with a Finite Volume Code*. DFVLR–FB 87–41, September 1987.

[64] N. Kroll and R. Radespiel. *An Improved Flux Vector Split Discretization Scheme for Viscous Flows*. DLR–FB 93–53, December 1993.

[65] N. Kroll, R. Radespiel, and C.-C. Rossow. *Accurate and Efficient Flow Solvers for 3D Applications on Structured Meshes*. In *Special Course on Parallel Computing in CFD*, number 807 in AGARD–R, chapter 4. AGARD — Advisory Group for Aerospace Research & Development, Neuilly-sur-Seine, France, October 1995.

[66] N. Kroll, C.-C. Rossow, K. Becker, and F. Thiele. MEGAFLOW — a numerical flow simulation system, 1998. 21st ICAS Congress, Melbourne, paper 98-2.7.4.

[67] N. Kroll, C.-C. Rossow, K. Becker, and F. Thiele. The MEGAFLOW project. *Aerospace Science and Technology*, 4:223–237, 2000.

[68] N. Kroll, C.-C. Rossow, D. Schwamborn, K. Becker, and G. Heller. MEGAFLOW — a numerical flow simulation tool for transport aircraft design, 2002. 23rd ICAS Congress, Toronto, paper 2002-1.10.5.

[69] R. F. Kunz and B. Lakshminarayana. Stability of explicit Navier–Stokes procedures using k–ε and k–ε/algebraic Reynolds stress turbulence models. *Journal of Computational Physics*, 103:141–159, 1992.

[70] P. D. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Communications on Pure and Applied Mathematics*, 7:159–193, 1954.

[71] B. van Leer, C.-H. Tai, and K. G. Powell. Design of optimally smoothing multi–stage schemes for the Euler equations, 1989. *AIAA paper* 89–1933–CP.

[72] M.-S. Liou and C. J. Jr. Steffen. A new flux splitting scheme. *Journal of Computational Physics*, 107:23–39, 1993.

[73] F. Liu and X. Zheng. A strongly coupled time–marching method for solving the Navier–Stokes and k–ω turbulence model equations with multigrid. *Journal of Computational Physics*, 128:289–300, 1996.

[74] L. Lorentzen and I. A. A. Lindblad. Application of two–equation and EARSM turbulence models to high lift aerodynamics, 1999. *AIAA paper* 99–3181.

[75] H. Luo, J. D. Baum, and R. Löhner. Computation of compressible flows using a two–equation turbulence model on unstructured grids, 1997. *AIAA paper* 97–0430.

[76] R. W. MacCormack. A new implicit algorithm for fluid flow, 1997. *AIAA paper* 97–2100.

[77] R. W. MacCormack. Iterative modified approximate factorization. *Computers & Fluids*, 30:917–925, 2001.

[78] R. W. MacCormack and T. Pulliam. Assessment of a new numerical procedure for fluid dynamics, 1998. *AIAA paper* 98–2821.

[79] J. C. Mandal and H. S. Rajput. An improved multigrid method for Euler equations. *Computational Mechanics*, 23:397–403, 1999.

[80] D. J. Mavriplis and L. Martinelli. *Multigrid Solution of Compressible Turbulent Flow on Unstructured Meshes using a Two–Equation Model*. ICASE Report 91–11, January 1991.

[81] D. J. Mavriplis and S. Pirzadeh. Large–scale parallel unstructured mesh computations for 3d high–lift analysis, 1999. *AIAA paper* 99–0537.

[82] MEGAFLOW — numerical flow simulation for aircraft design. CD ROM, June 2001. Project presentation.

[83] MEGAFLOW — numerical flow simulation for complete aircraft. CD ROM, December 2002. Closing Presentation DLR Project MEGAFLOW, December 10–11, 2002.

[84] F. R. Menter. Performance of popular turbulence models for attached and separated adverse pressure gradient flows, 1991. *AIAA paper* 91–1784.

[85] F. R. Menter. Zonal two equation k–ω turbulence models for aerodynamic flows, 1993. *AIAA paper* 93–2906.

[86] F. R. Menter. Untersuchung des Konvergenzverhaltens des CFD–Verfahrens FLOWer des DLR. Technical Report AEAT/TR–01–09, AEA Technology GmbH, October 2001.

[87] B. Merci, J. Steelant, J. Vierendeels, K. Riemslagh, and E. Dick. Computational treatment of source terms in two–equation turbulence models. *AIAA Journal*, 38(11):2085–2093, 2000.

[88] E. Monsen, M. Franke, T. Rung, P. Aumann, and A. Ronzheimer. Assessment of advanced transport–equation turbulence models for aircraft aerodynamic performance prediction, 1999. *AIAA paper* 99–3701.

[89] M. Moulton and J. Steinhoff. A technique for the simulation of stall with coarse–grid CFD methods, 2000. *AIAA paper* 2000–0277.

[90] W. Nitsche, H.-J. Heinemann, and R. Hilbig, editors. *New Results in Numerical and Experimental Fluid Mechanics II — Contributions to the 11th AG STAB/DGLR Symposium Berlin, Germany 1998*, volume 72 of *Notes on Numerical Fluid Mechanics*, Braunschweig, 1999. Vieweg Verlag.

[91] ONERA. *Manuel Théorique elsA*, July 2000. /ELSA/STB-97020/V1.4.

[92] D. W. Peaceman and H. H. Rachford. The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for Industrial and Applied Mathematics*, 3(1):28–41, 1955.

[93] J. Raddatz, DLR (Institute of Aerodynamics and Flow Technology), Braunschweig, Germany. private communications.

[94] R. Radespiel. *A Cell–Vertex Multigrid Method for the Navier–Stokes Equations*. NASA Technical Memorandum 101557 (NASA Langley), January 1989.

[95] R. Radespiel and N. Kroll. *Multigrid Schemes with Semicoarsening for Accurate Computations of Hypersonic Viscous Flows*. DLR–IB 129–90/19, April 1991.

[96] R. Radespiel and N. Kroll. *Extension of the Navier–Stokes Code CEVCATS to Hypersonic Equilibrium Flows*. DLR–IB 129–92/21, October 1992.

[97] R. Radespiel, C.-C. Rossow, and R. C. Swanson. Efficient cell–vertex multigrid scheme for the three–dimensional Navier–Stokes equations. *AIAA Journal*, 28(8):1464–1472, 1990.

[98] R. Radespiel and R. C. Swanson. Progress with multigrid schemes for hypersonic flow problems. *Journal of Computational Physics*, 116:103–122, 1995.

[99] M. Rakowitz, M. Sutcliffe, B. Eisfeld, D. Schwamborn, H. Bleecke, and J. Fassbender. Structured and unstructured computations on the DLR–F4 wing–body configuration, 2002. *AIAA paper* 2002–0837.

[100] A. Reusken. Fourier analysis of a robust multigrid method for convection–diffusion equations. *Numerische Mathematik*, 71:365–397, 1995.

[101] O. Reynolds. On the dynamical theory of incompressible viscous flows and the determination of the criterion. *Philosophical Transactions of the Royal Society of London*, 186(I):123–164, 1895.

[102] C.-C. Rossow. *Berechnung von Strömungsfeldern durch Lösung der Euler–Gleichungen mit einer erweiterten Finite–Volumen Diskretisierungsmethode (Calculations of Flow Fields by the Solution of the Euler Equations Using an Extended Finite Volume Discretization Scheme)*. Doctoral thesis, Technische Universität Braunschweig, 1989. (Published as DLR research report DLR–FB 89–38).

[103] C.-C. Rossow, O. Brodersen, and A. Ronzheimer. Grundlagen zur Erzeugung strukturierter Rechennetze. Beitrag zum Thema "Gittererzeugung und Datenstrukturen", S. 3-34, DMV-GAMM-GI Fachgruppe Numerische Software, 1997.

[104] C.-C. Rossow, DLR (Institute of Aerodynamics and Flow Technology), Braunschweig, Germany. private communications.

[105] R. Rudnik. *Untersuchung der Leistungsfähigkeit von Zweigleichungs–Turbulenzmodellen bei Profilumströmungen (Evaluation of the Performance of Two–Equation Turbulence Models for Airfoil Flows)*. Doctoral thesis, Technische Universität Berlin, 1997. (Published as DLR research report DLR–FB 97–49).

[106] R. Rudnik, A. Ronzheimer, and J. Raddatz. Numerical flow simulation for a wing/fuselage transport aircraft configuration with deployed high–lift system. In Nitsche et al. [90], pages 363–370.

[107] R. Rudnik, DLR (Institute of Aerodynamics and Flow Technology), Braunschweig, Germany. private communications.

[108] H. Schlichting and K. Gersten. *Grenzschicht–Theorie*. Springer–Verlag, Berlin, 9th edition, 1997.

[109] S. G. Sheffer, L. Martinelli, and A. Jameson. An efficient multigrid algorithm for compressible reactive flows. *Journal of Computational Physics*, 144:484–516, 1998.

[110] J. W. Sloof and W. Schmidt, editors. *Computational Aerodynamics Based on the Euler Equations*. Number 325 in AGARDograph. AGARD — Advisory Group for Aerospace Research & Development, Neuilly–sur–Seine, France, September 1994.

[111] P. R. Spalart and S. R. Allmaras. A one–equation turbulence model for aerodynamic flows, 1992. *AIAA paper* 92–0439.

[112] E. Sterner. A multigrid smoother for high Reynolds number flows. *Electronic Transactions on Numerical Analysis*, 6:234–245, 1997.

[113] H. W. Stock and W. Haase. Determination of length scales in algebraic turbulence models for Navier–Stokes methods. *AIAA Journal*, 27(1):5–14, 1989.

[114] K. Stüben and U. Trottenberg. Multigrid methods: Fundamental algorithms, model problem analysis and applications. In Hackbusch and Trottenberg [43], pages 1–176.

[115] R. C. Swanson and E. Turkel. *Multistage Scheme with Multigrid for Euler and Navier–Stokes Equations (Components and Analysis)*, August 1997. NASA Technical Paper 3631 (NASA Langley).

[116] A. Tassa, E. H. Atta, and L. A. Lemmerman. A new three-dimensional boundary layer calculation method, 1982. *AIAA paper* 82–0224.

[117] R. Temam. *Navier–Stokes Equations and Nonlinear Functional Analysis*, volume 66 of *CBMS–NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, 2nd edition, 1995.

[118] U. Trottenberg, C. Osterlee, and A. Schüller. *Multigrid*. Academic Press, San Diego, 2001.

[119] E. Truckenbrodt. *Fluidmechanik*, volume 1. Springer–Verlag, Berlin, 1980.

[120] E. Turkel, R. C. Swanson, V. N. Vatsa, and J. A. White. Multigrid for hypersonic viscous two– and three–dimensional flows, 1991. *AIAA paper* 91–1572–CP.

[121] E. Turkel and V. N. Vatsa. Effect of artificial viscosity on three–dimensional flow solutions. *AIAA Journal*, 32(1):39–45, 1994.

[122] E. Turkel, V. N. Vatsa, and V. Venkatakrishnan. Uni–directional implicit acceleration techniques for compressible Navier–Stokes solvers, 1999. *AIAA paper* 99–3265.

[123] R. S. Varga. *Matrix iterative analysis*. Prentice Hall, Englewood Cliff, New Jersey, 1962.

[124] V. N. Vatsa, M. D. Sanetrik, and E. B. Parlette. *A Multigrid–based multiblock flow solver for practical aerodynamic configurations*. In M. M. Hafez and D. A. Caughey, editors, *Frontiers of Computational Fluid Dynamics 1994*, pages 413–448. J. Wiley & Sons, Chichester, December 1994.

[125] M. E. Vázquez-Cendón. Improved treatment of source terms in upwind schemes for the shallow water equations in channels with irregular geometry. *Journal of Computational Physics*, 148:497–526, 1999.

[126] V. Venkatakrishnan. *Implicit Schemes and Parallel Computing in Unstructured Grid CFD*. ICASE Report 95–28, April 1995.

[127] V. Venkatakrishnan. Improved convergence of compressible Navier–Stokes solvers, 1998. *AIAA paper* 98–2967.

[128] R. Vichnevetsky and J. B. Bowles. *Fourier Analysis of Numerical Approximations of Hyperbolic Equations*. SIAM — Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1982.

[129] P. Wesseling. A survey of Fourier smoothing analysis results. In W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods III — Proceedings, Köln, 1991*, volume 98 of *International Series of Numerical Mathematics*, pages 105–127, Basel, 1991. Birkhäuser Verlag.

[130] P. Wesseling. *Introduction to Multigrid Methods*. ICASE Report 95–11, February 1995.

[131] D. C. Wilcox. Reassessment of the scale-determination equation for advanced turbulence models. *AIAA Journal*, 26(11):1299–1310, 1988. (*AIAA paper* 84–0176).

[132] D. C. Wilcox. *Turbulence Modelling for CFD*. DCW Industries, Inc., La Cañada, California, 1993.

[133] J. Zhu and T.-H. Shih. *CMOTT Turbulence Modul for NPARC*, August 1997. ICOMP–97–10 (CMOTT–97–05, NASA Lewis).

[134] Z. W. Zhu and C. Hirsch. Multigrid Fourier analysis of aspect ratio influence for Navier–Stokes computations, 1997. *AIAA paper* 97–1950.